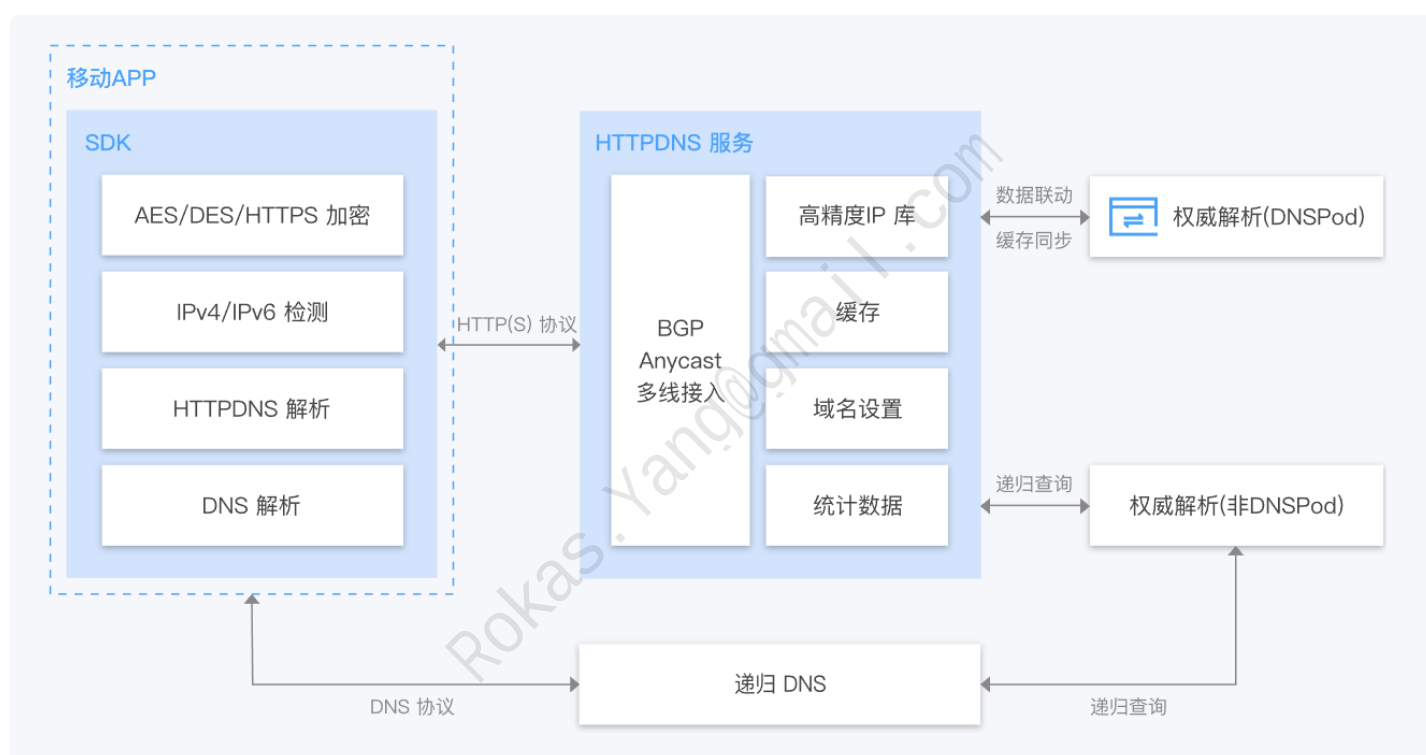


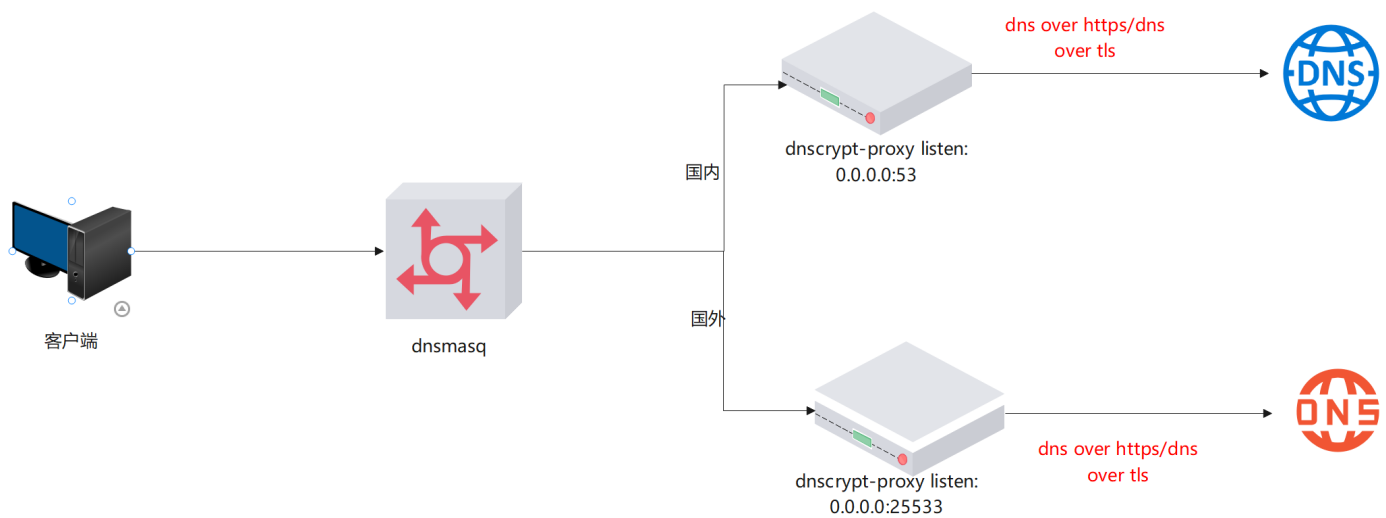
前言

普通DNS没有加密能力，所有查询默认都是UDP明文传输，当然有些返回字节结果过大也会截断采用TCP传输(retrying in TCP mode)，但这些都是明文；对于一些安全要求较高的业务场景，为避免出现劫持、污染等引起的安全威胁，DNS over HTTPS (DoH) 以及 DNS over TLS (DoT) 就派上了用场。同时HTTPDNS也可以规避运营商劫持的问题，主要原理就是绕过ISP提供的LDNS直接请求HTTPDNS服务，具有域名防劫持、调度精准等特性，但主要面向的是移动APP场景，解析流程如下：



回到正题，dnscrypt-proxy作为DoH/DoT的DNS转发服务，配合业界已经存在的各大DoH/DoT公共服务，轻松实现DNS加密传输。

在此基础上，又需要做到国内外域名分流走不同的DoH/DoT解析，以提高解析效率及精准度，这里使用了 `dnsmasq` + `dnscrypt-proxy` 实现，架构如下：



dnsmasq的安装配置这里不会详细展开介绍，可以参照[上篇文章](#)。

一、安装dnscrypt-proxy

1.软件源安装

各个发行版的软件仓库基本都会内置dnscrypt-proxy，也优先推荐选择此安装方式，会自动写好systemd服务。

发行版	安装命令
Arch	pacman -Sy dnscrypt-proxy/yay -Sy dnscrypt-proxy
CentOS/RedHat	yum install -y dnscrypt-proxy
Debian/Ubuntu	apt-get install -y dnscrypt-proxy

2. 二进制安装

(1) 获取

到 [releases](https://github.com/DNSCrypt/dnscrypt-proxy/releases) 页面下载最新版:

```
cd /opt #习惯放到/opt目录,你也可以放到/etc目录或其它目录下
wget https://github.com/DNSCrypt/dnscrypt-proxy/releases/download/2.1.2/dnscrypt-proxy-linux_x86_64-2.1.2.tar.gz
```

截至2022年11月最新版为2.1.2。

(2) 解压及软链

```
tar xvf dnscrypt-proxy-linux_x86_64-2.1.2.tar.gz
mv linux-x86_64 dnscrypt-proxy
cd dnscrypt-proxy
ln -sf /opt/dnscrypt-proxy/dnscrypt-proxy /usr/sbin/dnscrypt-proxy
```

- 重命名提升可读性,解压后,你会在目录下看到一个可执行的dnscrypt-proxy文件以及一些示例配置文件。
- 将可执行文件软链到PATH目录。

(3) 写systemd服务

```
vim /etc/systemd/system/dnscrypt-proxy.service
```

写入如下内容:

```
[Unit]
Description=Encrypted/authenticated DNS proxy
```

```
ConditionFileIsExecutable=/usr/sbin/dnscrypt-proxy

[Service]
StartLimitInterval=5
StartLimitBurst=10
ExecStart=/usr/sbin/dnscrypt-proxy "-config" "dnscrypt-proxy.toml"

WorkingDirectory=/opt/dnscrypt-proxy

Restart=always

RestartSec=120
EnvironmentFile=-/etc/sysconfig/dnscrypt-proxy

[Install]
WantedBy=multi-user.target
```

重载systemd守护进程:



```
systemctl daemon-reload
```

开机自启动:



```
systemctl enable dnscrypt-proxy.service
```

二、Public Servers与服务配置重写

1.临时测试解析情况

目录里面的 `example-dnscrypt-proxy.toml` 则为主要配置文件，把它拷贝一份作为正式配置文件:

```
cp example-dnscrypt-proxy.toml dnscrypt-proxy.toml
```

在还没有通过systemctl启动时可以先前台运行：

```
./dnscrypt-proxy "-config" "dnscrypt-proxy.toml"
```

测试解析是否正常：

```
./dnscrypt-proxy -resolve google.com
```

```
o 23:37:19 /opt/dnscrypt-proxy ./dnscrypt-proxy -config dnscrypt-proxy.toml
[2022-11-14 23:37:28] [NOTICE] dnscrypt-proxy 2.1.2
[2022-11-14 23:37:28] [NOTICE] Network connectivity detected
[2022-11-14 23:37:28] [NOTICE] Now listening to 127.0.0.1:53 [UDP]
[2022-11-14 23:37:28] [NOTICE] Now listening to 127.0.0.1:53 [TCP]
[2022-11-14 23:37:28] [NOTICE] Source [public-resolvers] loaded
[2022-11-14 23:37:28] [NOTICE] Source [relays] loaded
[2022-11-14 23:37:28] [NOTICE] Firefox workaround initialized
[2022-11-14 23:37:33] [NOTICE] [altername] TIMEOUT
[2022-11-14 23:37:34] [NOTICE] [techsaviours.org-dnscrypt] OK (DNSCrypt) - rtt: 208ms
[2022-11-14 23:37:35] [NOTICE] [doh-crypto-sx] OK (DoH) - rtt: 477ms
[2022-11-14 23:37:46] [NOTICE] [dct-del] OK (DNSCrypt) - rtt: 558ms
[2022-11-14 23:37:46] [NOTICE] [deffer-dns.au] OK (DNSCrypt) - rtt: 153ms
[2022-11-14 23:37:47] [NOTICE] [cloudflare] OK (DoH) - rtt: 169ms
[2022-11-14 23:37:48] [NOTICE] [dct-ru1] OK (DNSCrypt) - rtt: 192ms
[2022-11-14 23:37:58] [NOTICE] [dnscrypt.uk-ipv4] OK (DNSCrypt) - rtt: 361ms
[2022-11-14 23:37:58] [NOTICE] [starrydns] OK (DNSCrypt) - rtt: 51ms
[2022-11-14 23:37:58] [NOTICE] [dct-at1] OK (DNSCrypt) - rtt: 226ms
[2022-11-14 23:38:00] [CRITICAL] [meganerd-doh-ipv4] Certificate hash [72696d1113206db8c50dc557e562ee863af1972787bdb7da5b8f4240c3cc30bb] not found

Xeon-Debian11(dnscrypt&qbit) (1) x
o 23:37:31 /opt/dnscrypt-proxy ./dnscrypt-proxy -resolve google.com
Resolving [google.com] using 127.0.0.1 port 53

Resolver      : 89.58.6.169 (techsaviours.org.)
Canonical name: google.com.

IPv4 addresses: 216.58.214.78
IPv6 addresses: 2404:6800:4003:c02::66, 2404:6800:4003:c02::71, 2404:6800:4003:c02::65, 2404:6800:4003:c02::64

Name servers  : ns2.google.com., ns1.google.com., ns3.google.com., ns4.google.com.
DNSSEC signed : no
Mail servers  : 1 mail servers found

HTTPS alias   : -
HTTPS info    : [alpn]=[h2,h3]

Host info     : -
TXT records   : docusign=05958488-4752-4ef2-95eb-aa7ba8a3bd0e, docusign=1b0a6754-49b1-4db5-8540-d2c12664b289, v=spf1 include: spf.google.com ~all, google-site-verification=TV9-DBe4R80X4v0M4U_bd_J9cp0JM0nikft0jAgjmsQ, apple-domain-verification=30afiBcvSuDV2PLX, MS=E4A68B9AB28B9670BCE15412F62916164C0B208B, google-site-verification=wD8N1lJTNTkezJ49swvWw48f8_9xveREV4oB-0Hf5o, onetrust-domain-verification=de0led21f2fa4d8781cbc3ffb89cf4ef, atlassian-domain-verification=5YjTmWmjI92ewqkx2oXmBaD60Td9zWon9reakvHX6B77zzkF0to8PQ9QsKnbf4I, facebook-domain-verification=22rm551cu4k0ab0bxsw536tlds4h95, globalsecurity-smime-dv=CDYX+XFHwUw2wml6/Gb8+59B8sH31KzUr6c1l2BPvqKX8=, webexdomain-verification.8YX6G=6e6922db-e3e6-4a36-904e-a805c28087fa

o 23:37:42 /opt/dnscrypt-proxy
```

可以正常解析到地址，通过日志看，默认会请求内置的DoH server，因为这里DoH server大多是境外的，如果你在国内，此时科学上网能力是基础保障。

2.DoH/DoT服务列表

(1) public servers

dnscrypt-proxy官方提供一个[服务列表](#) (境内可能需要科学上网才能打开)，可以任意在支持的列表中选择上游DNS使用的服务器。

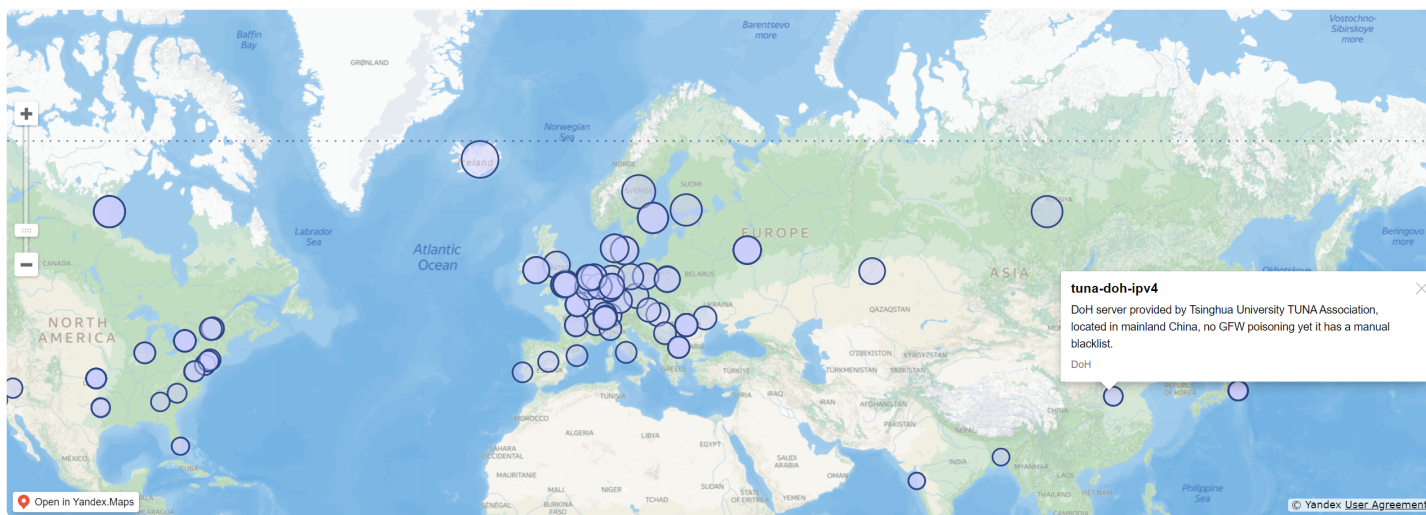
国内支持的目前只有阿里和清华大学的DoH server:

adguard-dns-family-ipv6	AdGuard DNS with safesearch and adult content blocking (over IPv6) Warning: This server is incompatible with anonymization.	DNSCrypt	
adguard-dns-ipv6	Remove ads and protect your computer from malware (over IPv6) Warning: This server is incompatible with anonymization.	DNSCrypt	
adguard-dns-unfiltered	AdGuard public DNS servers without filters Warning: This server is incompatible with anonymization.	DNSCrypt	
adguard-dns-unfiltered-doh	AdGuard public DNS servers without filters (over DoH)	DoH	
adguard-dns-unfiltered-ipv6	AdGuard public DNS servers without filters (over IPv6) Warning: This server is incompatible with anonymization.	DNSCrypt	
ahadns-doh-chi	A zero logging DNS with support for DNS-over-HTTPS (DoH) & DNS-over-TLS (DoT). Blocks ads, malware, trackers, viruses, ransomware, telemetry and more. No persistent logs. DNSSEC. Hosted in Chicago, USA. By https://ahadns.com/ Server statistics can be seen at: https://statistics.ahadns.com/?server=chi	DoH	
ahadns-doh-in	A zero logging DNS with support for DNS-over-HTTPS (DoH) & DNS-over-TLS (DoT). Blocks ads, malware, trackers, viruses, ransomware, telemetry and more. No persistent logs. DNSSEC. Hosted in Mumbai, India. By https://ahadns.com/ Server statistics can be seen at: https://statistics.ahadns.com/?server=in	DoH	
ahadns-doh-la	A zero logging DNS with support for DNS-over-HTTPS (DoH) & DNS-over-TLS (DoT). Blocks ads, malware, trackers, viruses, ransomware, telemetry and more. No persistent logs. DNSSEC. Hosted in Los Angeles, USA. By https://ahadns.com/ Server statistics can be seen at: https://statistics.ahadns.com/?server=la	DoH	
ahadns-doh-nl			
ahadns-doh-ny			
ahadns-doh-pl			
alidns-doh	alidns-doh A public DNS resolver that supports DoH/DoT in mainland China, provided by Alibaba-Cloud. Warning: GFW filtering rules are applied by that resolver. Homepage: https://alidns.com/ Protocol DoH Addresses['dns.alidns.com', '223.5.5.5'] Ports [443] DNSSEC false No filters false No logs false Stamp sdns://AgAAAAAAAAAACTiyMy41LjUuNSAUZf-XFWHwyJWwNPWQzxE3VDwpSDoT4pSfpwaLoFrgA5kbnMuYWxpZG5zLmNvbQovZG5zLXZlLXZlZXJ5		
altername			
altername-ipv6			
ams-ads-doh-nl			
ams-dnscrypt-nl	Resolver in Amsterdam. Dnscrypt protocol. Non-logging, non-filtering, DNSSEC.	DNSCrypt	
ams-dnscrypt-nl-ipv6	Resolver in Amsterdam. Dnscrypt protocol. Non-logging, non-filtering, DNSSEC.	DNSCrypt	
ams-doh-nl	Resolver in Amsterdam. DoH protocol. Non-logging, non-filtering, DNSSEC.	DoH	
ams-doh-nl-ipv6	Resolver in Amsterdam. DoH protocol. Non-logging, non-filtering, DNSSEC.	DoH	

sth-dnscrypt-se	Resolver in Stockholm, Sweden. DNSCrypt server. Non-logging, non-filtering, DNSSEC.	DNSCrypt	
sth-dnscrypt-se-ipv6	Resolver in Stockholm, Sweden. DNSCrypt server. Non-logging, non-filtering, DNSSEC.	DNSCrypt	
sth-doh-se	Resolver in Stockholm, Sweden. DoH server. Non-logging, non-filtering, DNSSEC.	DoH	
switch	Public DoH service provided by SWITCH in Switzerland https://www.switch.ch Provides protection against malware, but does not block ads.	DoH	
techsavours.org-dnscrypt	No filter No logs DNSSEC Nuremberg, Germany (netcup) Maintained by https://techsavours.org/	DNSCrypt	
tuna-doh-ipv4	DoH server provided by Tsinghua University TUNA Association, located in mainland China, no GFW poisoning yet it has a manual blacklist.	DoH	
uncensoreddns-dk-ipv4	Also known as censurfridns. DoH, no logs, no filter, DNSSEC, unicast hosted in Denmark - https://blog.uncensoreddns.org	DoH	
uncensoreddns-dk-ipv6	Also known as censurfridns. DoH, no logs, no filter, DNSSEC, unicast hosted in Denmark - https://blog.uncensoreddns.org	DoH	
uncensoreddns-ipv4	Also known as censurfridns. DoH, no logs, no filter, DNSSEC, anycast - https://blog.uncensoreddns.org	DoH	
uncensoreddns-ipv6	Also known as censurfridns. DoH, no logs, no filter, DNSSEC, anycast - https://blog.uncensoreddns.org	DoH	

(2) public servers map

同时还提供地图查找法，支持的点会在地图上标注出来：



方便选择距离你最近的DoH上游服务器。

3.配置文件重写

基于国内外分流需求，需要写两个配置文件，一个用于国外，一个用于国内。

(1) 国内配置

`dnscrypt-proxy.toml` 内置的大部分功能是我们不需要的，这里将 `dnscrypt-proxy.toml` 重写为：

```
# Empty listen_addresses to use systemd socket activation
listen_addresses = ['0.0.0.0:5533']
server_names = ['alidns-doh', 'tuna-doh-ipv4']
cache_size = 4096
cache_min_ttl = 2400
cache_max_ttl = 86400
cache_neg_min_ttl = 60
cache_neg_max_ttl = 600
[query_log]
    file = '/var/log/dnscrypt-proxy/query.log'

[nx_log]
    file = '/var/log/dnscrypt-proxy/nx.log'
```

```
[sources]
[sources.'public-resolvers']
url = 'https://download.dnscrypt.info/resolvers-list/v2/public-resolvers.md'
cache_file = '/var/cache/dnscrypt-proxy/public-resolvers.md'
minisign_key =
'RWQf6LRCGA9i53mLYecO4IzT51TGPpvWucNSch1CBM0QTaLn73Y7GF03'
refresh_delay = 72
prefix = ''
```

- 监听地址这里设置5533而非53，防止系统有内置的DNS服务，产生冲突；
- `server_names` 即为定义的上游DoH/DoT服务器，不能任意写，需要在支持的[服务列表](#)里面。

(2) 国外配置

把 `dnscrypt-proxy.toml` 拷贝一份：

```
cp dnscrypt-proxy.toml dnscrypt-proxy-foreign.toml
```

将 `server_names` 改成国外的DoH/DoT服务器，并修改监听端口，修改后的示例如下，其他地方不用删改：

```
listen_addresses = ['0.0.0.0:25533']
server_names = ['cloudflare', 'google']
```

4. 写systemd服务

国内配置已经写到systemd，那么国外配置同理，也需要一个进程监听运行。



```
vim /etc/systemd/system/dnscrypt-proxy-foreign.service
```

写入如下内容:



```
[Unit]
Description=Encrypted/authenticated DNS proxy
ConditionFileIsExecutable=/usr/sbin/dnscrypt-proxy

[Service]
StartLimitInterval=5
StartLimitBurst=10
ExecStart=/usr/sbin/dnscrypt-proxy "-config" "dnscrypt-proxy-foreign.toml"

WorkingDirectory=/opt/dnscrypt-proxy

Restart=always

RestartSec=120
EnvironmentFile=-/etc/sysconfig/dnscrypt-proxy

[Install]
WantedBy=multi-user.target
```

重载systemd守护进程:



```
systemctl daemon-reload
```

开机自启动:



```
systemctl enable dnscrypt-proxy-foreign.service
```

三、启动与测试验证

1.启动

国内外分别加载一个toml配置文件，并且已经写好对应的systemd服务了，接下来启动并测试下是否符合预期。

```
systemctl start dnscrypt-proxy.service
systemctl start dnscrypt-proxy-foreign.service
```

```
○ 23:43:11 在 /opt/dnscrypt-proxy 中 systemctl status dnscrypt-proxy.service
● dnscrypt-proxy.service - Encrypted/authenticated DNS proxy
   Loaded: loaded (/etc/systemd/system/dnscrypt-proxy.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2022-11-14 23:41:48 CST; 1min 54s ago
   TriggeredBy: ● dnscrypt-proxy.socket
   Main PID: 421122 (dnscrypt-proxy)
     Tasks: 12 (limit: 9508)
    Memory: 13.1M
       CPU: 180ms
    CGroup: /system.slice/dnscrypt-proxy.service
           └─421122 /usr/sbin/dnscrypt-proxy -config dnscrypt-proxy.toml

Nov 14 23:41:48 Debian dnscrypt-proxy[421122]: [2022-11-14 23:41:48] [NOTICE] Wiring systemd UDP socket #1, dnscrypt-proxy.socket, 127.0.2.1:53
Nov 14 23:41:48 Debian dnscrypt-proxy[421122]: [2022-11-14 23:41:48] [NOTICE] Source [public-resolvers] loaded
Nov 14 23:41:48 Debian dnscrypt-proxy[421122]: [2022-11-14 23:41:48] [NOTICE] Firefox workaround initialized
Nov 14 23:41:48 Debian dnscrypt-proxy[421122]: [2022-11-14 23:41:48] [NOTICE] [tuna-doh-ipv4] OK (DoH) - rtt: 39ms
Nov 14 23:41:48 Debian dnscrypt-proxy[421122]: [2022-11-14 23:41:48] [NOTICE] [alidns-doh] OK (DoH) - rtt: 15ms
Nov 14 23:41:48 Debian dnscrypt-proxy[421122]: [2022-11-14 23:41:48] [NOTICE] Sorted latencies:
Nov 14 23:41:48 Debian dnscrypt-proxy[421122]: [2022-11-14 23:41:48] [NOTICE] - 15ms alidns-doh
Nov 14 23:41:48 Debian dnscrypt-proxy[421122]: [2022-11-14 23:41:48] [NOTICE] - 39ms tuna-doh-ipv4
Nov 14 23:41:48 Debian dnscrypt-proxy[421122]: [2022-11-14 23:41:48] [NOTICE] Server with the lowest initial latency: alidns-doh (rtt: 15ms)
Nov 14 23:41:48 Debian dnscrypt-proxy[421122]: [2022-11-14 23:41:48] [NOTICE] dnscrypt-proxy is ready - live servers: 2
○ 23:43:42 在 /opt/dnscrypt-proxy 中 systemctl status dnscrypt-proxy-foreign.service
● dnscrypt-proxy-foreign.service - DNSCrypt client proxy
   Loaded: loaded (/lib/systemd/system/dnscrypt-proxy-foreign.service; disabled; vendor preset: enabled)
   Active: active (running) since Mon 2022-11-14 23:41:53 CST; 1min 55s ago
     Docs: https://github.com/DNSCrypt/dnscrypt-proxy/wiki
   Main PID: 421194 (dnscrypt-proxy)
     Tasks: 11 (limit: 9508)
    Memory: 12.6M
       CPU: 189ms
    CGroup: /system.slice/dnscrypt-proxy-foreign.service
           └─421194 /usr/sbin/dnscrypt-proxy -config /etc/dnscrypt-proxy/dnscrypt-proxy-foreign.toml

Nov 14 23:41:53 Debian dnscrypt-proxy[421194]: [2022-11-14 23:41:53] [NOTICE] Now listening to 0.0.0.0:25533 [TCP]
Nov 14 23:41:53 Debian dnscrypt-proxy[421194]: [2022-11-14 23:41:53] [NOTICE] Source [public-resolvers] loaded
Nov 14 23:41:53 Debian dnscrypt-proxy[421194]: [2022-11-14 23:41:53] [NOTICE] Firefox workaround initialized
Nov 14 23:41:53 Debian dnscrypt-proxy[421194]: [2022-11-14 23:41:53] [NOTICE] [google] OK (DoH) - rtt: 142ms
Nov 14 23:41:54 Debian dnscrypt-proxy[421194]: [2022-11-14 23:41:54] [NOTICE] [cloudflare] OK (DoH) - rtt: 161ms
Nov 14 23:41:54 Debian dnscrypt-proxy[421194]: [2022-11-14 23:41:54] [NOTICE] Sorted latencies:
Nov 14 23:41:54 Debian dnscrypt-proxy[421194]: [2022-11-14 23:41:54] [NOTICE] - 142ms google
Nov 14 23:41:54 Debian dnscrypt-proxy[421194]: [2022-11-14 23:41:54] [NOTICE] - 161ms cloudflare
Nov 14 23:41:54 Debian dnscrypt-proxy[421194]: [2022-11-14 23:41:54] [NOTICE] Server with the lowest initial latency: google (rtt: 142ms)
Nov 14 23:41:54 Debian dnscrypt-proxy[421194]: [2022-11-14 23:41:54] [NOTICE] dnscrypt-proxy is ready - live servers: 2
○ 23:43:48 在 /opt/dnscrypt-proxy 中 lsof -i :5533
COMMAND      PID USER   FD   TYPE    DEVICE  SIZE/OFF  NODE NAME
dnscrypt- 421122 root    8u    IPv6 25981542      0t0  UDP *:5533
dnscrypt- 421122 root    9u    IPv6 25981543      0t0  TCP *:5533 (LISTEN)
○ 23:43:51 在 /opt/dnscrypt-proxy 中 lsof -i :25533
COMMAND      PID USER   FD   TYPE    DEVICE  SIZE/OFF  NODE NAME
dnscrypt- 421194 dnscrypt-proxy 6u    IPv6 25976414      0t0  UDP *:25533
dnscrypt- 421194 dnscrypt-proxy 7u    IPv6 25976415      0t0  TCP *:25533 (LISTEN)
○ 23:43:53 在 /opt/dnscrypt-proxy 中
```

服务正常监听并运行，5533监听用于国内域名，25533监听用于国外域名。

2.测试验证

此时并没有做国内外智能分流，先单独验证下两个服务是否都正常解析。

查看解析日志文件：

```
tail -f /var/log/dnscrypt-proxy/query.log
```

dig命令测试不同端口的解析所走的上游DNS:

```
dig qcloud.com @192.168.1.72 -p 5533 +short
dig youtube.com @192.168.1.72 -p 25533 +short
```

```
23:50:47 /opt/dnscrypt-proxy tail -f /var/log/dnscrypt-proxy/query.log
[2022-11-14 23:50:56] 192.168.1.16 qcloud.com A PASS 41ms alidns-doh 1 走阿里doh
[2022-11-14 23:51:11] 192.168.1.16 youtube.com A PASS 520ms cloudflare 2 走cf doh
[2022-11-14 23:51:15] 192.168.1.16 youtube.com A PASS 0ms
[2022-11-14 23:51:18] 192.168.1.16 qcloud.com A PASS 0ms 3 重复查询命中缓存
```

```
Xeon-Internal-Kali x
(root@kali) - [~]
# dig qcloud.com @192.168.1.72 -p 5533 +short
119.29.42.201
119.29.47.192

(root@kali) - [~]
# dig youtube.com @192.168.1.72 -p 25533 +short
142.250.189.174

(root@kali) - [~]
# dig youtube.com @192.168.1.72 -p 25533 +short
142.250.189.174

(root@kali) - [~]
# dig qcloud.com @192.168.1.72 -p 5533 +short
119.29.42.201
119.29.47.192

(root@kali) - [~]
#
```

到此，说明两个服务运行状态良好。

四、Dnsmasq实现国内外域名智能分流

1.修改dnsmasq上游DNS

如dnsmasq还没安装配置，可参考[上篇文章](#)，直到做到 `dnsmasq-china-list` 这一步实现dnsmasq维度的国内外分流。

之后，只需修改上游DNS为dnscrypt-proxy运行的机器即可，如果dnsmasq和dnscrypt-proxy在同一台机器运行，则修改成本地地址即可。

(1) 指定国内上游DoH监听地址

国内Doh则需修改 `dnsmasq-china-list` 里的 `accelerated-domains.china.conf`，将IP替换为国内DoH监听地址：

```
sed -i 's|114.114.114.114|127.0.0.1#5533|g' accelerated-domains.china.conf
```

替换后的结果形如 `server=/0-100.com/127.0.0.1#5533`

因为我这里将两个服务独立运行在两个系统，所以替换为对应运行dnscrypt-proxy的机器的国内DoH监听地址：

```
00:21:20 /etc/dnsmasq.d head accelerated-domains.china.conf
server=/0-100.com/192.168.1.72#5533
server=/0-6.com/192.168.1.72#5533
server=/00.net/192.168.1.72#5533
server=/000.link/192.168.1.72#5533
server=/00000.host/192.168.1.72#5533
server=/00042.com/192.168.1.72#5533
server=/00058.com/192.168.1.72#5533
server=/0006266.com/192.168.1.72#5533
server=/0007.net/192.168.1.72#5533
server=/000714.xyz/192.168.1.72#5533
00:24:36 /etc/dnsmasq.d
```

(2) 指定国外上游DoH监听地址

因为 `resolv.conf` 不能指定端口，需要在 `dnsmasq.conf` 定义上游DNS为国外DoH的监听地址。所以 `/etc/resolv.conf` 这里写本地地址，`dnsmasq.conf` 加一条到国外DoH监听地址的配置：

```
$ cat /etc/resolv.conf
nameserver 127.0.0.1
$ cat /etc/dnsmasq.conf
log-queries
log-facility=/var/log/dnsmasq.log
no-hosts
bogus-nxdomain=119.29.29.29
cache-size=1000
port=53
```

#以下为增加的配置，25533前面的#号并非注释，而是指定端口，如果dnsmasq和dnscrypt-proxy在同一台机器，替换为127.0.0.1#25533
server=192.168.1.72#25533

(3) iptables DNAT规则

基于[#\(2\)](#)的拓展方案，可选项，二者任选其一，适用于dnsmasq和dnscrypt-proxy在不同机器运行的情况。

如果不想在 `dnsmasq.conf` 里面指定server为国外DoH地址，而是直接在 `/etc/resolv.conf` 设置，那么只需在dnscrypt-proxy机器做一条53端口DNAT到国外DoH监听端口的规则：

```
#192.168.1.72 替换为dnscrypt-proxy机器内网IP
iptables -t nat -A PREROUTING -i ens192 -p tcp --dport 53 -j DNAT --
to-destination 192.168.1.72:25533
iptables -t nat -A PREROUTING -i ens192 -p udp --dport 53 -j DNAT --
to-destination 192.168.1.72:25533
```

```
00:42:49 /opt/dnscrypt-proxy iptables -t nat -A PREROUTING -i ens192 -p tcp --dport 53 -j DNAT --to-destination 192.168.1.72:25533
00:43:02 /opt/dnscrypt-proxy iptables -t nat -A PREROUTING -i ens192 -p udp --dport 53 -j DNAT --to-destination 192.168.1.72:25533
00:43:09 /opt/dnscrypt-proxy iptables -L -n -v -t nat

Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source    destination
      0      0 DNAT      tcp  --  ens192 *      0.0.0.0/0      0.0.0.0/0      tcp dpt:53 to:192.168.1.72:25533
      0      0 DNAT      udp  --  ens192 *      0.0.0.0/0      0.0.0.0/0      udp dpt:53 to:192.168.1.72:25533

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source    destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source    destination
  653 83781 DOCKER    all  --  *      *        0.0.0.0/0      !127.0.0.0/8      ADDRTYPE match dst-type LOCAL

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source    destination
      0      0 MASQUERADE all  --  *      *        !docker0 172.17.0.0/16    0.0.0.0/0

Chain DOCKER (1 references)
  pkts bytes target    prot opt in     out     source    destination
      0      0 RETURN    all  --  docker0 *        0.0.0.0/0      0.0.0.0/0
00:43:12 /opt/dnscrypt-proxy
```

此时修改dnsmasq机器的 `/etc/resolv.conf` 文件内容如下：

```
nameserver 192.168.1.72
```

2.验证DoH/DoT智能分流场景

使用任意一台内网机器，将DNS解析修改为dnsmasq的机器地址，之后访问国内外域名测试验证。

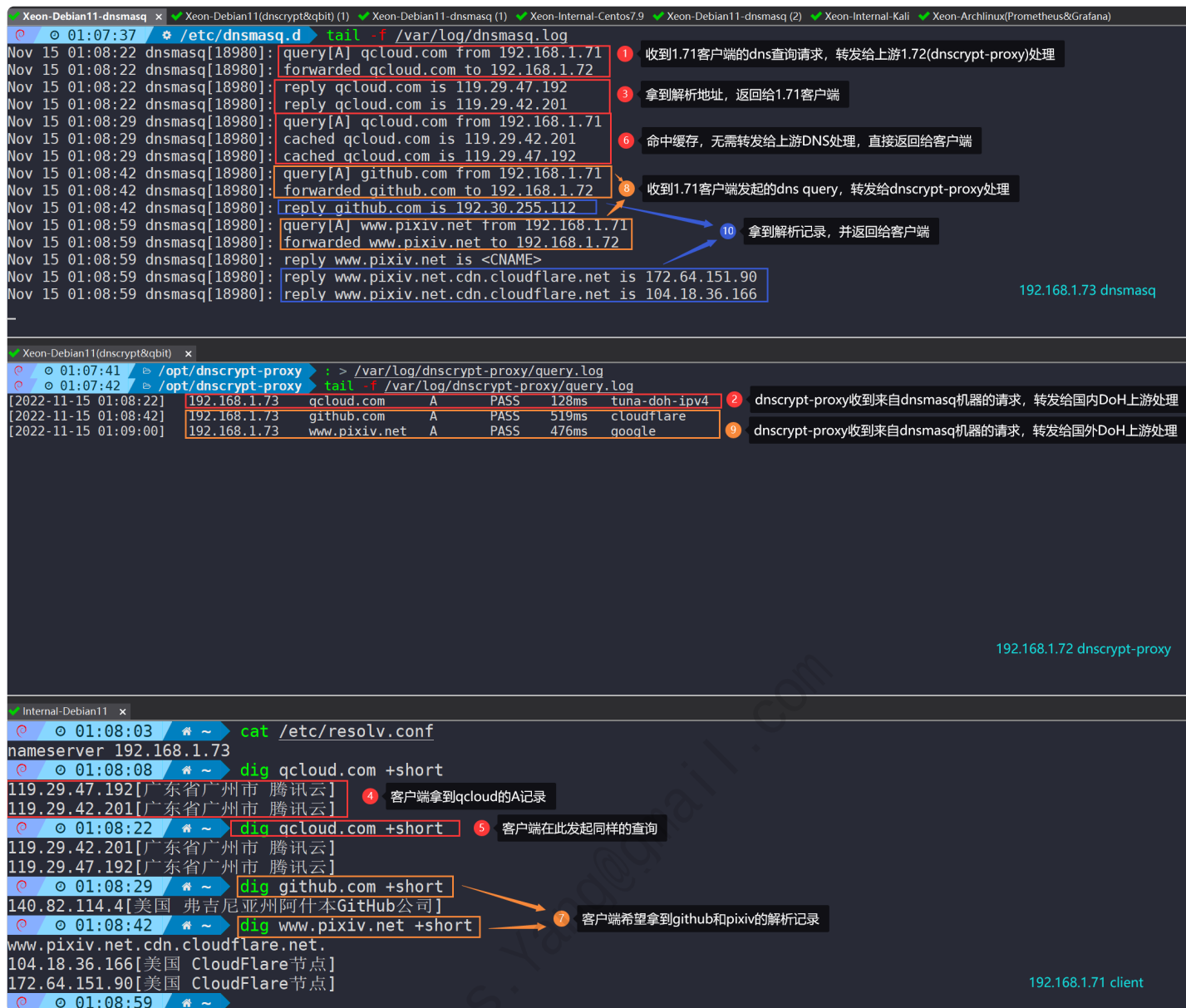
机器	角色/服务
192.168.1.71	客户端，dns指向dnsmasq
192.168.1.72	dnscrypt-proxy
192.168.1.73	dnsmasq

(1) 日志验证

如果你完全按照我上面的配置来搭建，那么对应的日志文件路径如下：

- dnsmasq query日志路径： `/var/log/dnsmasq.log`
- dnscrypt-proxy query日志路径： `/var/log/dnscrypt-proxy/query.log`

使用dig命里测试，并监听对应的日志文件输出：



可以清晰看到，国内域名解析，dnsmasq转发给dnscrypt-proxy的国内上游DoH，国外域名解析，则转发给国外上游DoH，并且缓存记录在dnsmasq做了控制，重复请求直接命中缓存返回给客户端。

(2) 抓包验证

`dig github.com` 作为验证示例：

```

01:48:49 /etc/dnsmasq.d tail -f /var/log/dnsmasq.log
Nov 15 01:48:14 dnsmasq[18980]: query[A] zhihu.com from 192.168.1.71
Nov 15 01:48:14 dnsmasq[18980]: forwarded zhihu.com to 192.168.1.72
Nov 15 01:48:14 dnsmasq[18980]: reply zhihu.com is 103.41.167.234
Nov 15 01:48:36 dnsmasq[18980]: query[A] qcloud.com from 192.168.1.71
Nov 15 01:48:36 dnsmasq[18980]: forwarded qcloud.com to 192.168.1.72
Nov 15 01:48:36 dnsmasq[18980]: reply qcloud.com is 119.29.47.192
Nov 15 01:48:36 dnsmasq[18980]: reply qcloud.com is 119.29.42.201
Nov 15 01:48:42 dnsmasq[18980]: query[A] github.com from 192.168.1.71
Nov 15 01:48:42 dnsmasq[18980]: forwarded github.com to 192.168.1.72
Nov 15 01:48:43 dnsmasq[18980]: reply github.com is 140.82.114.3

```

```

Xeon-Debian11(dnscrypt&qbit) x
/opt/dnscrypt-proxy tail -f /var/log/dnscrypt-proxy/query.log
[2022-11-15 01:08:22] 192.168.1.73 qcloud.com A PASS 128ms tuna-doh-ipv4
[2022-11-15 01:08:42] 192.168.1.73 github.com A PASS 519ms cloudflare
[2022-11-15 01:09:00] 192.168.1.73 www.pixiv.net A PASS 476ms google
[2022-11-15 01:48:14] 192.168.1.73 zhihu.com A PASS 138ms tuna-doh-ipv4
[2022-11-15 01:48:36] 192.168.1.73 qcloud.com A PASS 157ms tuna-doh-ipv4
[2022-11-15 01:48:43] 192.168.1.73 github.com A PASS 454ms google

```

```

Internal-Debian11 x
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 5988
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags::; udp: 1024
;; QUESTION SECTION:
;github.com.                IN      A
;; ANSWER SECTION:
github.com.                798     IN      A      140.82.114.4[美国 弗吉尼亚州阿什本GitHub公司]

```

同时在dnsmasq和dnscrypt-proxy机器上部署抓包:

dnsmasq机器:

```

tcpdump -i any -nn -s 0 port 53 or port 5533 or port 25533 or port 443 -v -w dnsmasq.pcap

```

dnscrypt-proxy机器:

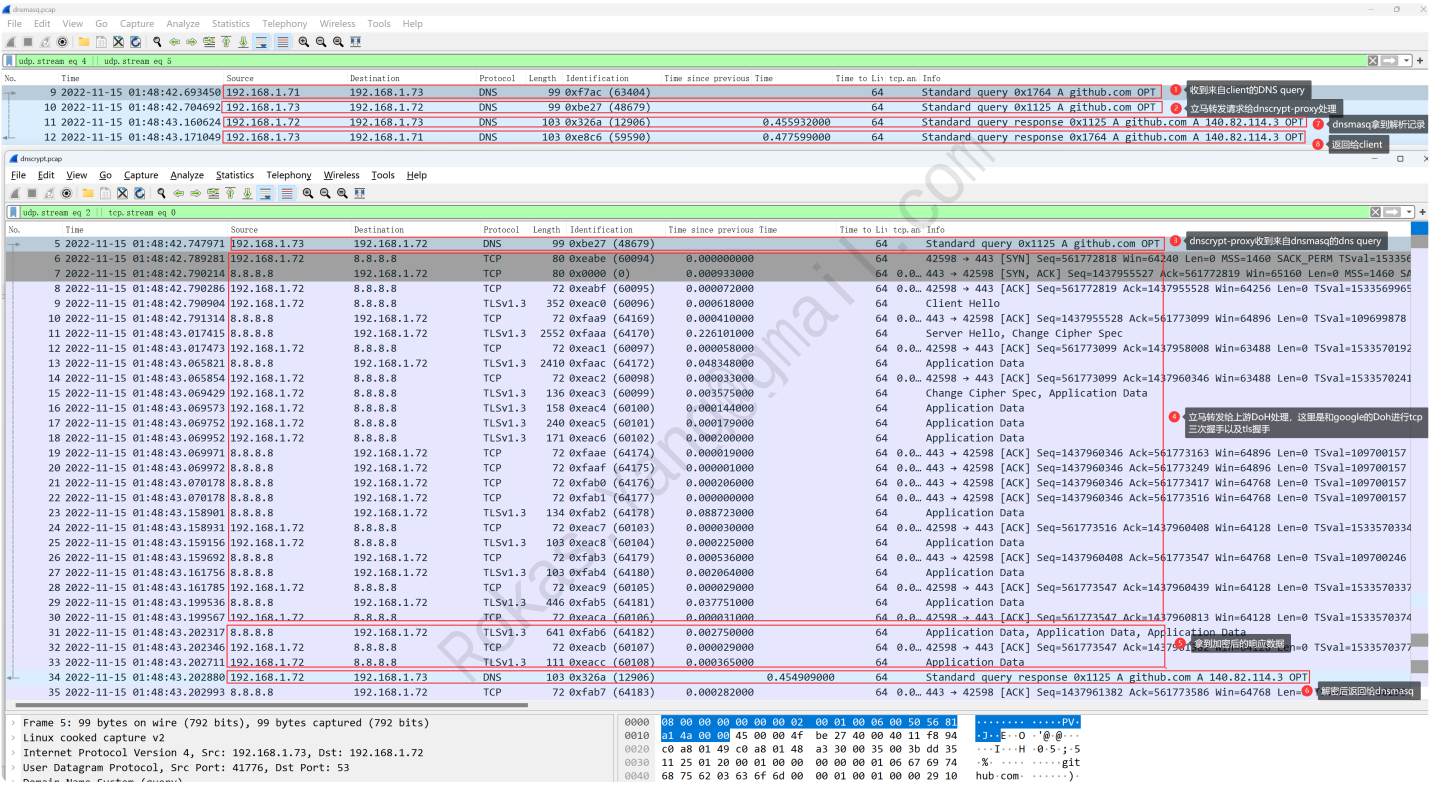
```

tcpdump -i any -nn -s 0 port 53 or port 5533 or port 25533 or port 443 -v -w dnscrypt-proxy.pcap

```

之后将 `.pcap` 文件下载到本地，使用wireshark分析如下：

- dnsmasq: 9号包，收到来自客户端的DNS query请求；
- dnsmasq: 10号包，dnsmasq向上游dnscrypt-proxy请求；
- dnscrypt-proxy: 5号包，dnscrypt-proxy收到来自dnsmasq的dns query请求；
- dnscrypt-proxy: 6-30号包，向上游Google DoH建立TCP三次握手以及TLS握手，加密传输DNS请求；
- dnscrypt-proxy: 31-33号包，拿到加密后的响应数据；
- dnscrypt-proxy: 34号包，解密后将解析结果返回给dnsmasq；
- dnsmasq: 11号包，拿到解析记录；
- dnsmasq: 12号包，将结果返回给客户端。



可以清晰看到，dnsmasq收到请求后，转发给国外DoH服务器处理，并且此过程经过了TLS加密传输，极大的保证了DNS安全性，拿到请求并且解密后正常返回给客户端。

总结

到此，`dnsmasq` + `dnscrypt-proxy` 实现国内外DoH/DoT分流解析已经全部完成。实现原理也很简单，dnsmasq机器作为入口，使用 `dnsmasq-china-list` 大陆域名白名单实现分流转发给上游dnscrypt-proxy处理，dnscrypt-proxy再往对应的DoH/DoT public servers加密转发拿到解析结果。

另外，dnscrypte-proxy还有负载均衡能力，在 `toml` 配置文件中通过 `lb_strategy` 参数指定，参数范围可以是：

- `first`：总是选择列表中最快的服务器
- `p2`：随机选择前2名最快的服务器，默认选项
- `ph`：在所有服务器中最快的一半之间随机选择
- `random`：从Server列表中随机选取

根据不同业务场景调整，同时可以多选几个优质上游DoH/DoT，适当增加dnscrypte-proxy的RS数量，提升优选对象。

Rokas.Yang@gmail.com