

1、详细描述客户端发起的每一次 DNS 请求和结果

答：.pcapng包中有关DNS的数据包如下：

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.1.1	DNS	77	Standard query 0x3eae A mirror.azure.cn
2	0.000193171	10.11.29.131	210.22.70.3	DNS	77	Standard query 0x799c A mirror.azure.cn
3	0.000214658	10.11.29.131	210.22.84.3	DNS	77	Standard query 0x799c A mirror.azure.cn
4	0.000237609	10.11.29.131	114.114.114.114	DNS	77	Standard query 0x799c A mirror.azure.cn
5	0.003621049	210.22.84.3	10.11.29.131	DNS	168	Standard query response 0x799c A mirror.azure.cn CNAME mirror.trafficmanager.cn CNAME eas...
6	0.003673107	210.22.70.3	10.11.29.131	DNS	168	Standard query response 0x799c A mirror.azure.cn CNAME mirror.trafficmanager.cn CNAME eas...
7	0.003748760	127.0.1.1	127.0.0.1	DNS	168	Standard query response 0x3eae A mirror.azure.cn CNAME mirror.trafficmanager.cn CNAME eas...
25	0.026855458	114.114.114.114	10.11.29.131	DNS	168	Standard query response 0x799c A mirror.azure.cn CNAME mirror.trafficmanager.cn CNAME eas...

- 1)、编号1：客户端的操作系统检查自己本地的hosts文件是否有要解析的dns网址映射关系
- 2)、编号2、3、4：客户端解析到ip为210.22.70.3、210.22.84.3、114.114.114.114的dns，并发送给服务端
- 3)、编号5、6：服务端解析到ip为210.22.84.3、210.22.70.3的dns，并把ip返回给客户端
- 4)、编号7：客户端的操作系统检查自己本地的hosts文件是否有ip为114.114.114.114的dns网址映射关系
- 5)、编号25：服务端解析到ip为114.114.114.114的dns，并把这个ip返回给客户端

2、说明客户端与服务端建立了多少个 TCP channel，分别是哪些 frame，分别完成了什么传输任务，为什么存在多个 TCP channel

答：

- 1) 5个
- 2) 以.pcapng包中的数据为例，包含的frame如下：

第一个（编号为8~10）：

8	0.004007447	10.11.29.131	139.217.146.62	TCP	76	60686 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1759700185 TSecr=0...
9	0.010020167	139.217.146.62	10.11.29.131	TCP	76	80 → 60686 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1440 SACK_PERM=1 TSval=3890471...
10	0.010069075	10.11.29.131	139.217.146.62	TCP	68	60686 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=1759700191 TSecr=3890471464

第二个（编号为27、37、38）：

27	0.034843260	10.11.29.131	139.217.146.62	TCP	76	60690 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1759700216 TSecr=0...
28	0.040718016	139.217.146.62	10.11.29.131	HTTP	307	HTTP/1.1 200 OK
29	0.040793723	139.217.146.62	10.11.29.131	HTTP	1516	Continuation
30	0.040807551	10.11.29.131	139.217.146.62	TCP	68	60686 → 80 [ACK] Seq=846 Ack=11161 Win=56448 Len=0 TSval=1759700222 TSecr=3890471472
31	0.040850489	139.217.146.62	10.11.29.131	HTTP	4412	ContinuationContinuation
32	0.040857300	10.11.29.131	139.217.146.62	TCP	68	60686 → 80 [ACK] Seq=846 Ack=15505 Win=65024 Len=0 TSval=1759700222 TSecr=3890471472
33	0.040968456	139.217.146.62	10.11.29.131	HTTP	1516	Continuation
34	0.041022212	139.217.146.62	10.11.29.131	HTTP	876	Continuation
35	0.041028857	10.11.29.131	139.217.146.62	TCP	68	60686 → 80 [ACK] Seq=846 Ack=17761 Win=70912 Len=0 TSval=1759700222 TSecr=3890471472
36	0.041734621	10.11.29.131	139.217.146.62	HTTP	479	GET /icons/kaiyuanshe.png HTTP/1.1
37	0.042309250	139.217.146.62	10.11.29.131	TCP	76	80 → 60690 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1440 SACK_PERM=1 TSval=3890471...
38	0.042337212	10.11.29.131	139.217.146.62	TCP	68	60690 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=1759700223 TSecr=3890471472

第三个（编号为97、112、113）， 第四个（编号为98、110、111）， 第五个（编号为99、115、116）：

97	0.094614156	10.11.29.131	139.217.146.62	TCP	76 60692 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1759700275 TSecr=0..
98	0.095078214	10.11.29.131	139.217.146.62	TCP	76 60694 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1759700276 TSecr=0..
99	0.095621726	10.11.29.131	139.217.146.62	TCP	76 60696 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1759700276 TSecr=0..
100	0.096490989	139.217.146.62	10.11.29.131	HTTP	305 HTTP/1.1 200 OK
101	0.096640097	139.217.146.62	10.11.29.131	HTTP	294 Continuation
102	0.097123987	10.11.29.131	139.217.146.62	TCP	68 60686 → 80 [ACK] Seq=1677 Ack=148851 Win=261248 Len=0 TSval=1759700278 TSecr=3890471..
103	0.097269162	139.217.146.62	10.11.29.131	TCP	68 80 → 60690 [ACK] Seq=1 Ack=415 Win=30080 Len=0 TSval=3890471486 TSecr=1759700271
104	0.097288194	139.217.146.62	10.11.29.131	HTTP	306 HTTP/1.1 200 OK
105	0.097301948	10.11.29.131	139.217.146.62	TCP	68 60690 → 80 [ACK] Seq=415 Ack=239 Win=30336 Len=0 TSval=1759700278 TSecr=3890471486
106	0.097456985	139.217.146.62	10.11.29.131	HTTP	479 Continuation
107	0.097475095	10.11.29.131	139.217.146.62	TCP	68 60690 → 80 [ACK] Seq=415 Ack=650 Win=31360 Len=0 TSval=1759700278 TSecr=3890471486
108	0.098709355	10.11.29.131	139.217.146.62	HTTP	497 GET /icons/mirror-setting-tips.png HTTP/1.1
109	0.098859672	10.11.29.131	139.217.146.62	HTTP	488 GET /icons/quick-link.png HTTP/1.1
110	0.101642481	139.217.146.62	10.11.29.131	TCP	76 80 → 60694 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1440 SACK_PERM=1 TSval=3890471..
111	0.101680113	10.11.29.131	139.217.146.62	TCP	68 60694 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=1759700282 TSecr=3890471487
112	0.101685351	139.217.146.62	10.11.29.131	TCP	76 80 → 60692 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1440 SACK_PERM=1 TSval=3890471..
113	0.101699575	10.11.29.131	139.217.146.62	TCP	68 60692 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=1759700282 TSecr=3890471487
114	0.101883557	10.11.29.131	139.217.146.62	HTTP	485 GET /icons/contact.png HTTP/1.1
115	0.102156541	139.217.146.62	10.11.29.131	TCP	76 80 → 60696 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1440 SACK_PERM=1 TSval=3890471..
116	0.102178580	10.11.29.131	139.217.146.62	TCP	68 60696 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=1759700283 TSecr=3890471487

3) 5个不同的客户端端口和服务端80端口通信

### 3、选择几个 frame 详细说明一次 TCP 握手流程，需要包含具体 frame 内容

答：以.pcapng包中编号为8~10的数据包为例，如下：

8	0.004007447	10.11.29.131	139.217.146.62	TCP	76 60686 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1759700185 TSecr=0..
9	0.010020167	139.217.146.62	10.11.29.131	TCP	76 80 → 60686 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1440 SACK_PERM=1 TSval=3890471..
10	0.010069075	10.11.29.131	139.217.146.62	TCP	68 60686 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=1759700191 TSecr=3890471464

- 1)、第一次握手：客户端（10.11.29.131:60686）将标志位SYN置为1，随机产生一个Seq为i = 0，向服务端（139.217.146.62:80）发送一个不带payload(tcp.len = 0)的数据包，客户端进入SYN\_SENT状态，等待服务端确认
- 2)、第二次握手：服务端收到数据包后由标志位SYN=1知道客户端请求建立连接，服务端将SYN和ACK都置为1，Ack = i + 1 = 1，随机产生一个Seq为j = 0，然后向客户端发送一个不带payload(tcp.len = 0)的数据包以确认连接请求，服务端进入SYN\_RCVD状态
- 3)、第三次握手：客户端收到确认后，检查Ack是否为i+1=1，ACK是否为1，如果正确则将标志位ACK置为1，Ack = j + 1 = 1,向服务端发送一个不带payload(tcp.len = 0)的数据包，服务端检查Ack是否为j+1 = 1，ACK是否为1，如果正确则连接建立成功，客户端和服务端进入ESTABLISHED状态，完成三次握手，随后客户端和服务端之间可以开始传送数据

### 4、请说明服务端程序可以如何优化，以提升单个用户访问延迟，以及并发吞吐量

答：可尝试如下方式：

- 1)、数据结构和算法优化
- 2)、数据库层：索引优化，慢查询消除（关系型），批量操作减少IO，通过引入NoSQL提升系统性能
- 3)、网络IO：批量操作，通过pipeline管道操作，减少数据库IO请求次数。
- 4)、缓存：使用内存数据库 redis / memcached。（内存读取时间远高于磁盘读取，使用内存数据库抗一些并发比较高的请求）
- 5)、异步：asyncio , celery (使用一些异步框架或者异步库)
- 6)、并发：gevent / 多线程（并发相关请求）