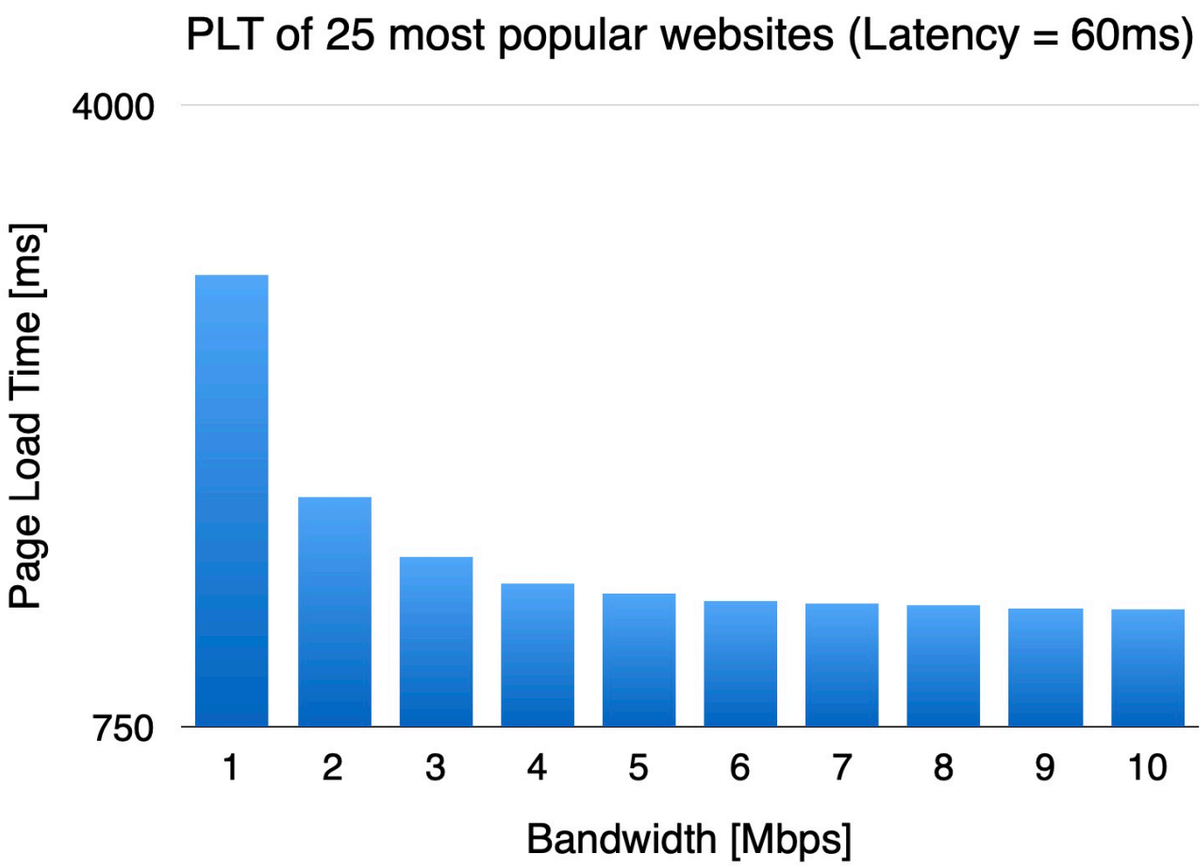
网络性能分析

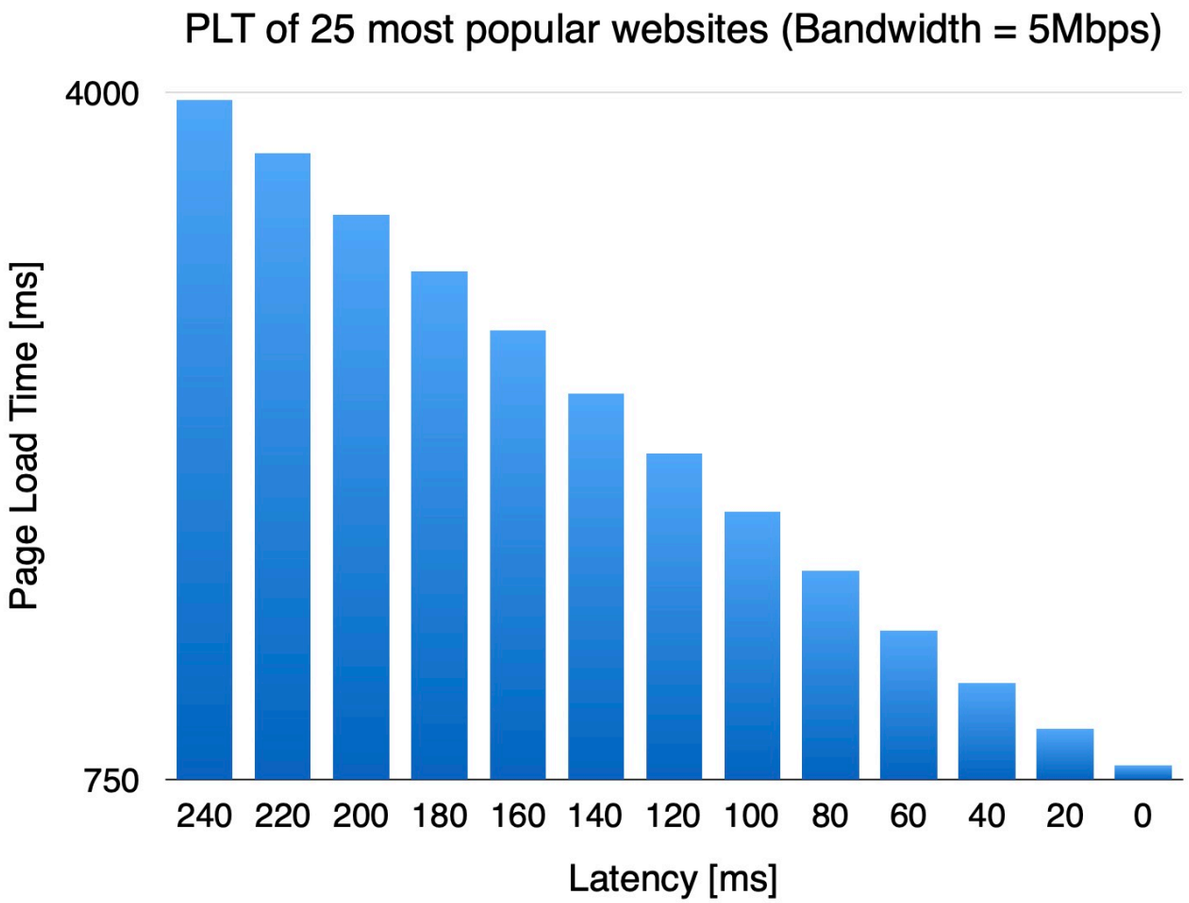
丁静 dingjingdjdj@gmail.com

性能瓶颈

- 延迟：从信号源到目的地所用的时间
  - ★ 传播延迟 消息从发送端到接收端需要的时间，是信号传播距离和速度的函数
  - ★ 传输延迟 消息中的比特转移到链路需要的时间，是消息长度和链路速率的函数
  - ★ 处理延迟
  - ★ 排队延迟
- 带宽：信道的最大吞吐量

延迟与带宽


PLT of 25 most popular websites (Latency = 60ms)
4000
750
Page Load Time [ms]
1
2
3
4
5
6
7
8
9
10
Bandwidth [Mbps]


PLT of 25 most popular websites (Bandwidth = 5Mbps)
4000
750
Page Load Time [ms]
240
220
200
180
160
140
120
100
80
60
40
20
0
Latency [ms]

TCP Header

TCP Header

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port														
4	32	Sequence number																														
8	64	Acknowledgment number (if ACK set)																														
12	96	Data offset			Reserved 0 0 0				NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size														
16	128	Checksum																Urgent pointer (if URG set)														
20 ...	160 ...	Options (if *data offset* > 5. Padded at the end with "0" bytes if necessary.) ...																														

可变窗口

- net.ipv4.tcp_window_scaling = 1

```
Window size value: 221
[Calculated window size: 28288]
[Window size scaling factor: 128]
```

- net.core.rmem_max,net.ipv4.tcp_rmem 取决于最大值
- 窗口数据大小 = 带宽 * RTT, 带宽延迟积 BDP

假设网络带宽为 1000Mb/s, RTT 为 1ms, 窗口大小 = 1000/8 * (1/1000) = 0.125M

- net.ipv4.tcp_adv_win_scale = 1

0.125M * 2 = 0.25M

- netstat -s | grep -E "pruned|collapsed"

  9433 packets pruned from receive queue because of socket buffer overrun

  433976 packets collapsed in receive queue due to low socket buffer

可变窗口

- net.core.rmem_default = 87380
- net.core.rmem_max = 212992
- net.ipv4.tcp_rmem = 4096 87380   6291456
- setsockopt(SO_RCVBUF) , setsockopt(SO_SNDBUF) 显式设置缓冲区

可变窗口

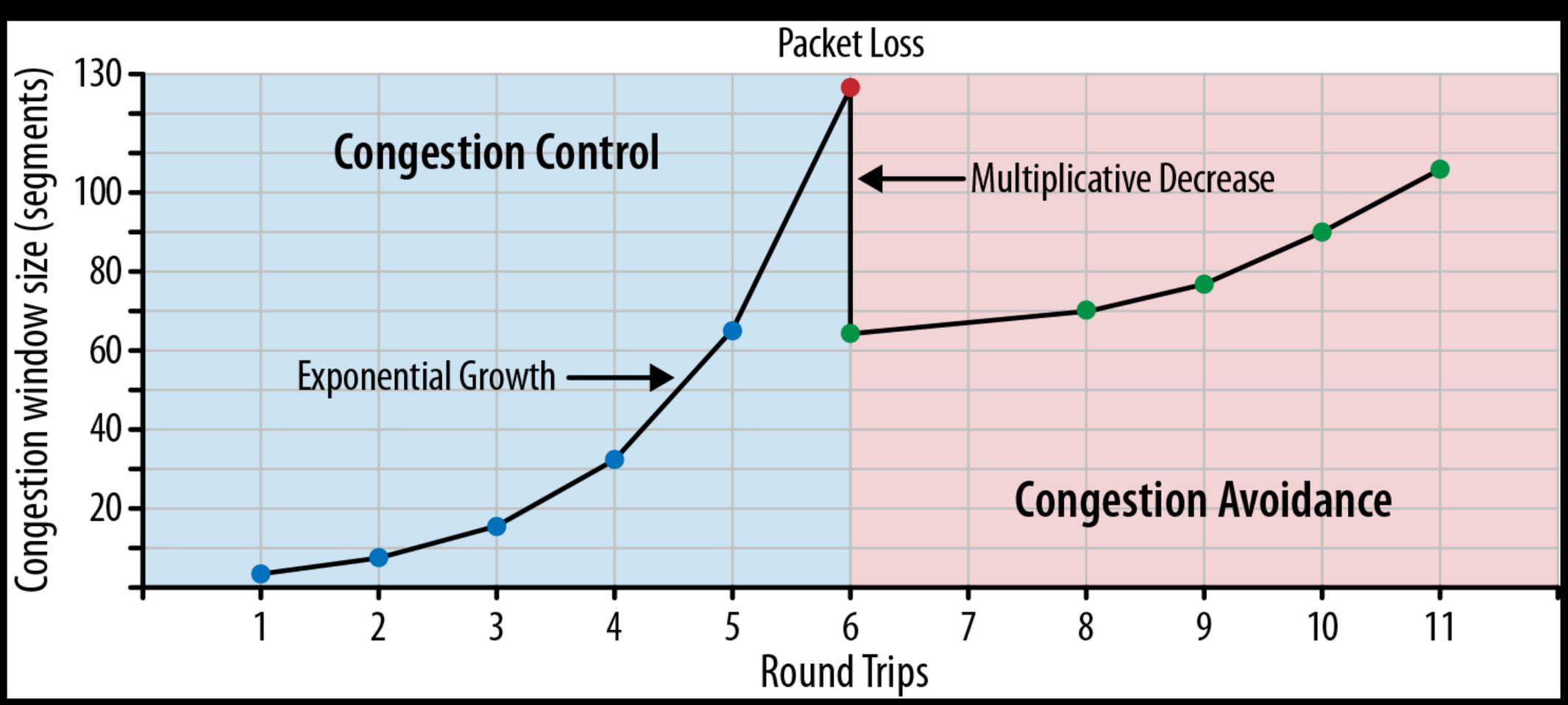
).	Time	Source	Destination	Protocol	Length	Frame	Time since previo	TCP Segment Ler	Info
6629	11:50:56.5…	10.0.0.8	10.0.0.211	TCP	74	Yes	0.000000000	0	38306 → 9028 [SYN] Seq=0 Win=13600 Len=0 MSS=1360 SACK_PERM=1 TSval=1445674456 TSecr=0
6630	11:50:56.5…	10.0.0.211	10.0.0.8	TCP	74	Yes	0.000015000	0	9028 → 38306 [SYN, ACK] Seq=0 Ack=1 Win=13480 Len=0 MSS=1360 SACK_PERM=1 TSval=73370745
6631	11:50:56.5…	10.0.0.8	10.0.0.211	TCP	66	Yes	0.000203000	0	38306 → 9028 [ACK] Seq=1 Ack=1 Win=13696 Len=0 TSval=1445674456 TSecr=73370745
6632	11:50:56.5…	10.0.0.8	10.0.0.211	TCP	5458	Yes	0.000428000	5392	38306 → 9028 [ACK] Seq=1 Ack=1 Win=13696 Len=5392 TSval=1445674456 TSecr=73370745
6633	11:50:56.5…	10.0.0.211	10.0.0.8	TCP	66	Yes	0.000035000	0	9028 → 38306 [ACK] Seq=1 Ack=5393 Win=24320 Len=0 TSval=73370746 TSecr=1445674456
6634	11:50:56.5…	10.0.0.8	10.0.0.211	TCP	8154	Yes	0.000030000	8088	38306 → 9028 [PSH, ACK] Seq=5393 Ack=1 Win=13696 Len=8088 TSval=1445674456 TSecr=733707
6635	11:50:56.5…	10.0.0.211	10.0.0.8	TCP	66	Yes	0.000019000	0	9028 → 38306 [ACK] Seq=1 Ack=13481 Win=34944 Len=0 TSval=73370746 TSecr=1445674456
6636	11:50:56.5…	10.0.0.8	10.0.0.211	TCP	2762	Yes	0.000159000	2696	38306 → 9028 [ACK] Seq=13481 Ack=1 Win=13696 Len=2696 TSval=1445674457 TSecr=73370746
6637	11:50:56.5…	10.0.0.211	10.0.0.8	TCP	66	Yes	0.000017000	0	9028 → 38306 [ACK] Seq=1 Ack=16177 Win=32640 Len=0 TSval=73370746 TSecr=1445674457
6638	11:50:56.5…	10.0.0.8	10.0.0.211	TCP	4110	Yes	0.000051000	4044	38306 → 9028 [PSH, ACK] Seq=16177 Ack=1 Win=13696 Len=4044 TSval=1445674457 TSecr=73370
6639	11:50:56.5…	10.0.0.211	10.0.0.8	TCP	66	Yes	0.000007000	0	9028 → 38306 [ACK] Seq=1 Ack=20221 Win=28672 Len=0 TSval=73370746 TSecr=1445674457
6640	11:50:56.5…	10.0.0.8	10.0.0.211	TCP	1414	Yes	0.000109000	1348	38306 → 9028 [ACK] Seq=20221 Ack=1 Win=13696 Len=1348 TSval=1445674457 TSecr=73370746
6641	11:50:56.5…	10.0.0.211	10.0.0.8	TCP	66	Yes	0.000012000	0	9028 → 38306 [ACK] Seq=1 Ack=21569 Win=27392 Len=0 TSval=73370746 TSecr=1445674457
6642	11:50:56.5…	10.0.0.8	10.0.0.211	TCP	5458	Yes	0.000018000	5392	38306 → 9028 [ACK] Seq=21569 Ack=1 Win=13696 Len=5392 TSval=1445674457 TSecr=73370746
6643	11:50:56.5…	10.0.0.211	10.0.0.8	TCP	66	Yes	0.000009000	0	9028 → 38306 [ACK] Seq=1 Ack=26961 Win=22016 Len=0 TSval=73370746 TSecr=1445674457
6645	11:50:56.5…	10.0.0.8	10.0.0.211	TCP	2762	Yes	0.000053000	2696	38306 → 9028 [ACK] Seq=26961 Ack=1 Win=13696 Len=2696 TSval=1445674457 TSecr=73370746
6647	11:50:56.5…	10.0.0.8	10.0.0.211	TCP	1414	Yes	0.000184000	1348	38306 → 9028 [ACK] Seq=29657 Ack=1 Win=13696 Len=1348 TSval=1445674457 TSecr=73370746
6649	11:50:56.5…	10.0.0.8	10.0.0.211	TCP	6806	Yes	0.000068000	6740	38306 → 9028 [PSH, ACK] Seq=31005 Ack=1 Win=13696 Len=6740 TSval=1445674457 TSecr=73370
6650	11:50:56.5…	10.0.0.8	10.0.0.211	TCP	4110	Yes	0.000007000	4044	38306 → 9028 [ACK] Seq=37745 Ack=1 Win=13696 Len=4044 TSval=1445674457 TSecr=73370746
6651	11:50:56.5…	10.0.0.8	10.0.0.211	TCP	6806	Yes	0.000050000	6740	38306 → 9028 [ACK] Seq=41789 Ack=1 Win=13696 Len=6740 TSval=1445674457 TSecr=73370746
6961	11:50:56.5…	10.0.0.211	10.0.0.8	TCP	66	Yes	0.039697000	0	9028 → 38306 [ACK] Seq=1 Ack=48529 Win=3840 Len=0 TSval=73370787 TSecr=1445674457
6964	11:50:56.5…	10.0.0.8	10.0.0.211	TCP	1414	Yes	0.000288000	1348	38306 → 9028 [ACK] Seq=48529 Ack=1 Win=13696 Len=1348 TSval=1445674497 TSecr=73370787
6966	11:50:56.5…	10.0.0.8	10.0.0.211	TCP	1414	Yes	0.000028000	1348	38306 → 9028 [ACK] Seq=49877 Ack=1 Win=13696 Len=1348 TSval=1445674497 TSecr=73370787
7484	11:50:56.6…	10.0.0.211	10.0.0.8	TCP	66	Yes	0.039677000	0	9028 → 38306 [ACK] Seq=1 Ack=51225 Win=1152 Len=0 TSval=73370827 TSecr=1445674497
11414	11:50:56.8…	10.0.0.8	10.0.0.211	TCP	1218	Yes	0.211915000	1152	[TCP Window Full] 38306 → 9028 [PSH, ACK] Seq=51225 Ack=1 Win=13696 Len=1152 TSval=1445
11416	11:50:56.8…	10.0.0.211	10.0.0.8	TCP	66	Yes	0.000014000	0	[TCP ZeroWindow] 9028 → 38306 [ACK] Seq=1 Ack=52377 Win=0 Len=0 TSval=73371038 TSecr=14
13851	11:50:57.0…	10.0.0.8	10.0.0.211	TCP	66	Yes	0.209801000	0	[TCP Keep-Alive] 38306 → 9028 [ACK] Seq=52376 Ack=1 Win=13696 Len=0 TSval=1445674959 TS
13852	11:50:57.0…	10.0.0.211	10.0.0.8	TCP	66	Yes	0.000017000	0	[TCP ZeroWindow] 9028 → 38306 [ACK] Seq=1 Ack=52377 Win=0 Len=0 TSval=73371248 TSecr=14
13902	11:50:57.0…	10.0.0.211	10.0.0.8	TCP	66	Yes	0.005161000	0	[TCP Window Update] 9028 → 38306 [ACK] Seq=1 Ack=52377 Win=40448 Len=0 TSval=73371253 T
13960	11:50:57.0…	10.0.0.8	10.0.0.211	TCP	1610	Yes	0.002148000	1544	38306 → 9028 [PSH, ACK] Seq=52377 Ack=1 Win=13696 Len=1544 TSval=1445674964 TSecr=73371
13961	11:50:57.0…	10.0.0.211	10.0.0.8	TCP	66	Yes	0.000033000	0	9028 → 38306 [ACK] Seq=1 Ack=53921 Win=42112 Len=0 TSval=73371256 TSecr=1445674964
13962	11:50:57.0…	10.0.0.8	10.0.0.211	TCP	2762	Yes	0.000016000	2696	38306 → 9028 [PSH, ACK] Seq=53921 Ack=1 Win=13696 Len=2696 TSval=1445674964 TSecr=73371
13963	11:50:57.0…	10.0.0.211	10.0.0.8	TCP	66	Yes	0.000020000	0	9028 → 38306 [ACK] Seq=1 Ack=56617 Win=39808 Len=0 TSval=73371256 TSecr=1445674964
13964	11:50:57.0…	10.0.0.8	10.0.0.211	TCP	1414	Yes	0.000004000	1348	38306 → 9028 [ACK] Seq=56617 Ack=1 Win=13696 Len=1348 TSval=1445674964 TSecr=73371253
13965	11:50:57.0…	10.0.0.211	10.0.0.8	TCP	66	Yes	0.000005000	0	9028 → 38306 [ACK] Seq=1 Ack=57965 Win=38528 Len=0 TSval=73371256 TSecr=1445674964
13966	11:50:57.0…	10.0.0.8	10.0.0.211	TCP	6806	Yes	0.000004000	6740	38306 → 9028 [ACK] Seq=57965 Ack=1 Win=13696 Len=6740 TSval=1445674964 TSecr=73371253
13967	11:50:57.0…	10.0.0.211	10.0.0.8	TCP	66	Yes	0.000015000	0	9028 → 38306 [ACK] Seq=1 Ack=64705 Win=55936 Len=0 TSval=73371256 TSecr=1445674964
13986	11:50:57.0…	10.0.0.8	10.0.0.211	TCP	1414	Yes	0.002568000	1348	38306 → 9028 [ACK] Seq=64705 Ack=1 Win=13696 Len=1348 TSval=1445674969 TSecr=73371256
13987	11:50:57.0…	10.0.0.211	10.0.0.8	TCP	66	Yes	0.000018000	0	9028 → 38306 [ACK] Seq=1 Ack=66053 Win=58624 Len=0 TSval=73371258 TSecr=1445674969

慢启动

- 拥塞窗口大小(cwnd)

发送端对从客户端接受确认ACK之前可以发送数据量的限制

- 拥塞避免



TCP Fast Open

- 内核3.7 linux， Android
-

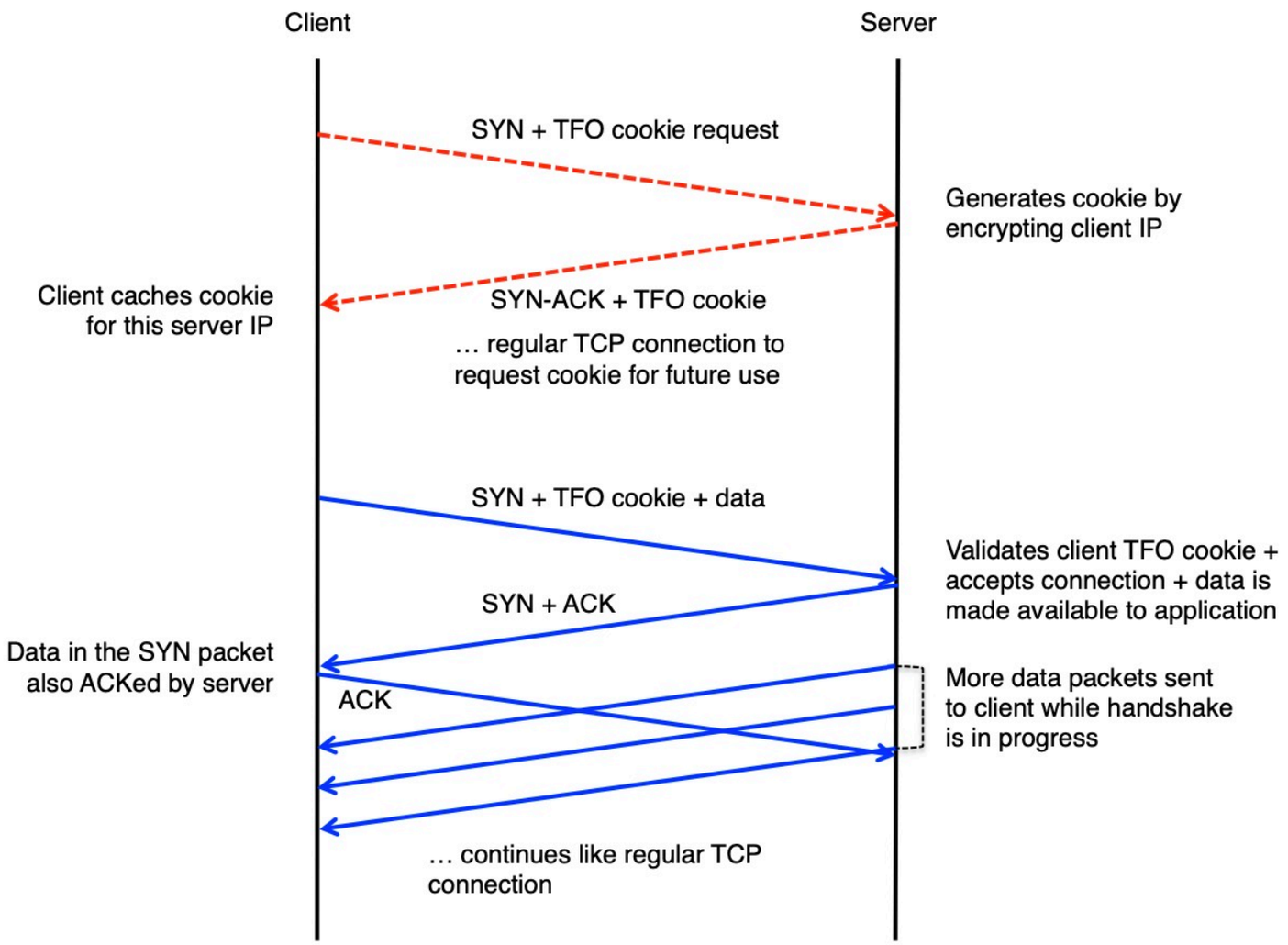


Figure 3: TFO connection overview

TCP Nagle 算法

- 减少大量小包的发送，在TCP连接上最多只能有一个未被确认的未完成的小分组，在该分组的确认到达之前不能发送其他小分组

- delay ack 40ms

- 内部网络交互关闭，redis，mysql，服务之间

- 关闭能减少延迟，但会增加数据量

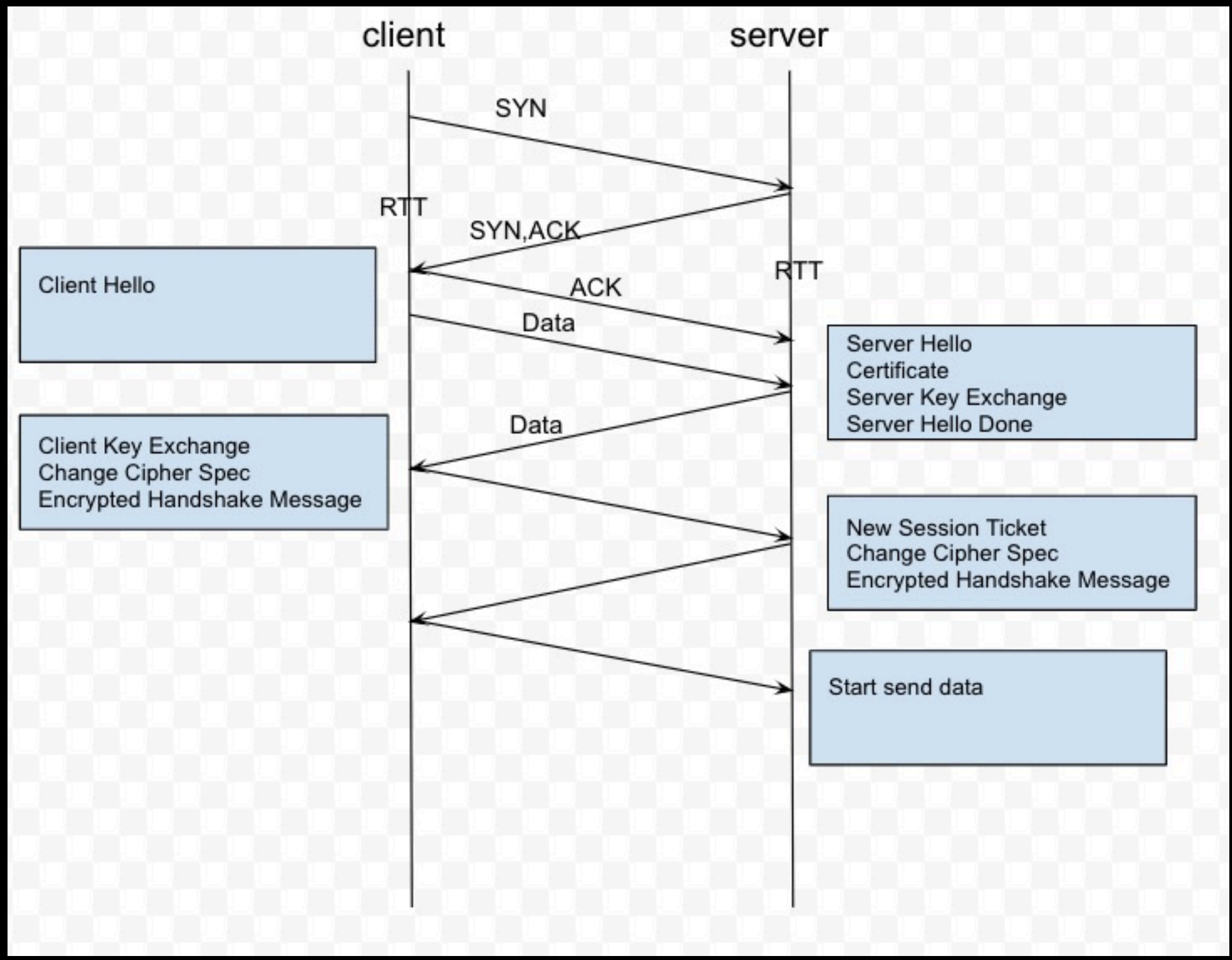
- https://cloud.tencent.com/developer/article/1004431

TCP 优化指南

- 更新内核，确保 initcwnd = 10
- 关闭慢启动重启 net.ipv4.tcp_slow_start_after_idle=0
- 某些场景下 TCP fast open
- 打开 net.ipv4.tcp_timestamps=1,关闭net.ipv4.tcp_tw_recycle=0
- 选择性确认 net.ipv4.tcp_sack=1
- 关闭 tcp nagle setsockopt(TCP_NODELAY)
- 启用窗口缩放 net.ipv4.tcp_window_scaling = 1
- 减少传输冗余数据，压缩数据传输
- 尽量重用链接

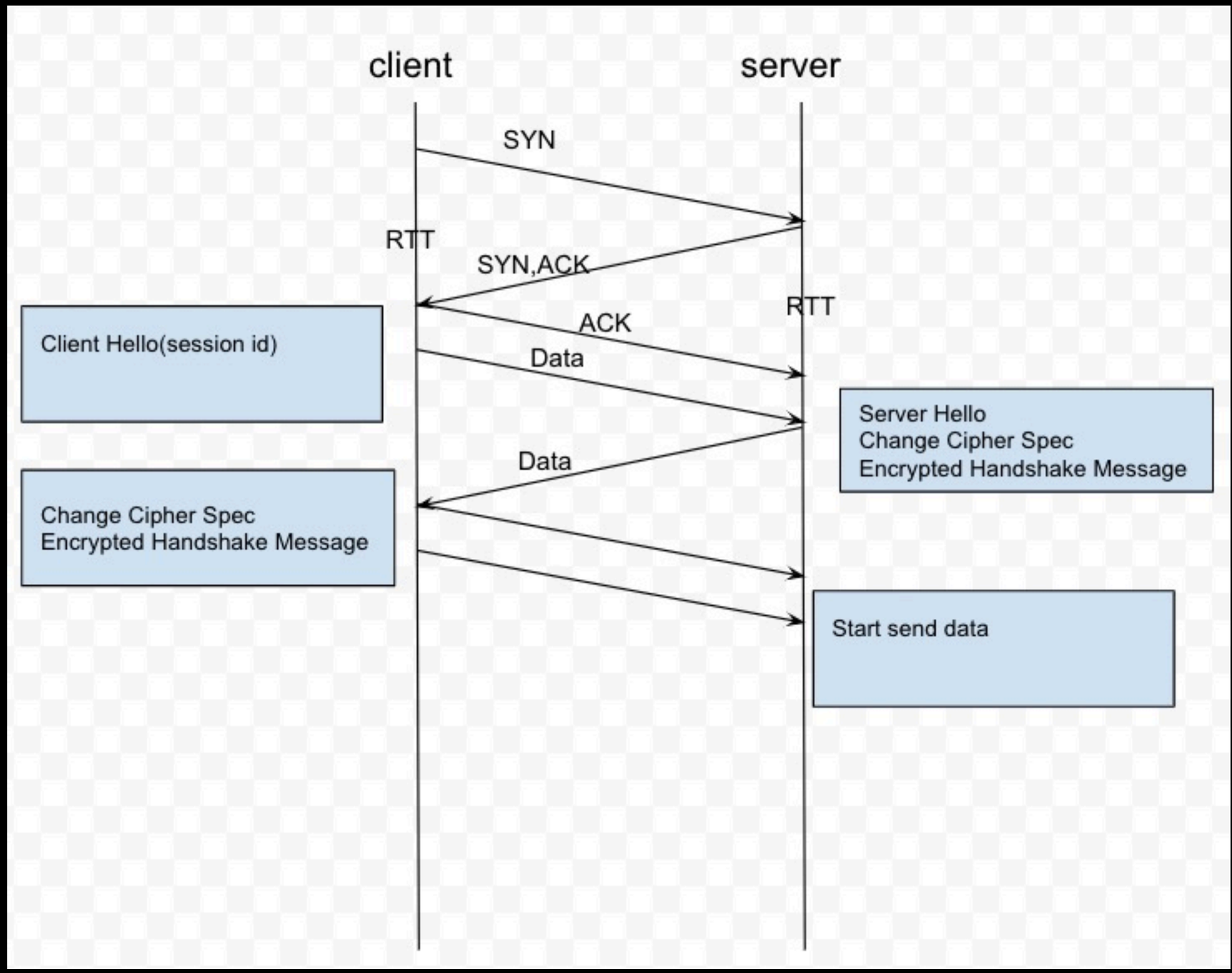
SSL

- 会话层协议


client
server
SYN
RTT
SYN,ACK
Client Hello
RTT
ACK
Data
Server Hello
Certificate
Server Key Exchange
Server Hello Done
Data
Client Key Exchange
Change Cipher Spec
Encrypted Handshake Message
New Session Ticket
Change Cipher Spec
Encrypted Handshake Message
Start send data

SSL

- TLS 会话恢复
- 会话记录单



SSL 优化指南

- TLS 库保证最新版本
- 启用并配置会话缓存和无状态恢复
- 在接近用户的地方完成 TLS 会话，尽量减少往返延迟
- 确保证书链不会超过拥塞窗口大小
- 禁用服务器的TLS压缩功能
- 配置 TLS 记录大小，能封装在一个 TCP 段内
- http://www.infoq.com/cn/presentations/performance-optimization-of-https(搜索 腾讯HTTPS性能优化实践)

HTTP

- 减少 DNS 查询
- 减少 HTTP 请求
- 使用CDN
- 文本 Gzip 压缩
- 使用缓存，Expires， Cache Control，Last-Modified, Etag
- 合并 CSS， JS 文件
- 嵌入小资源
- 减少重定向请求

HTTP

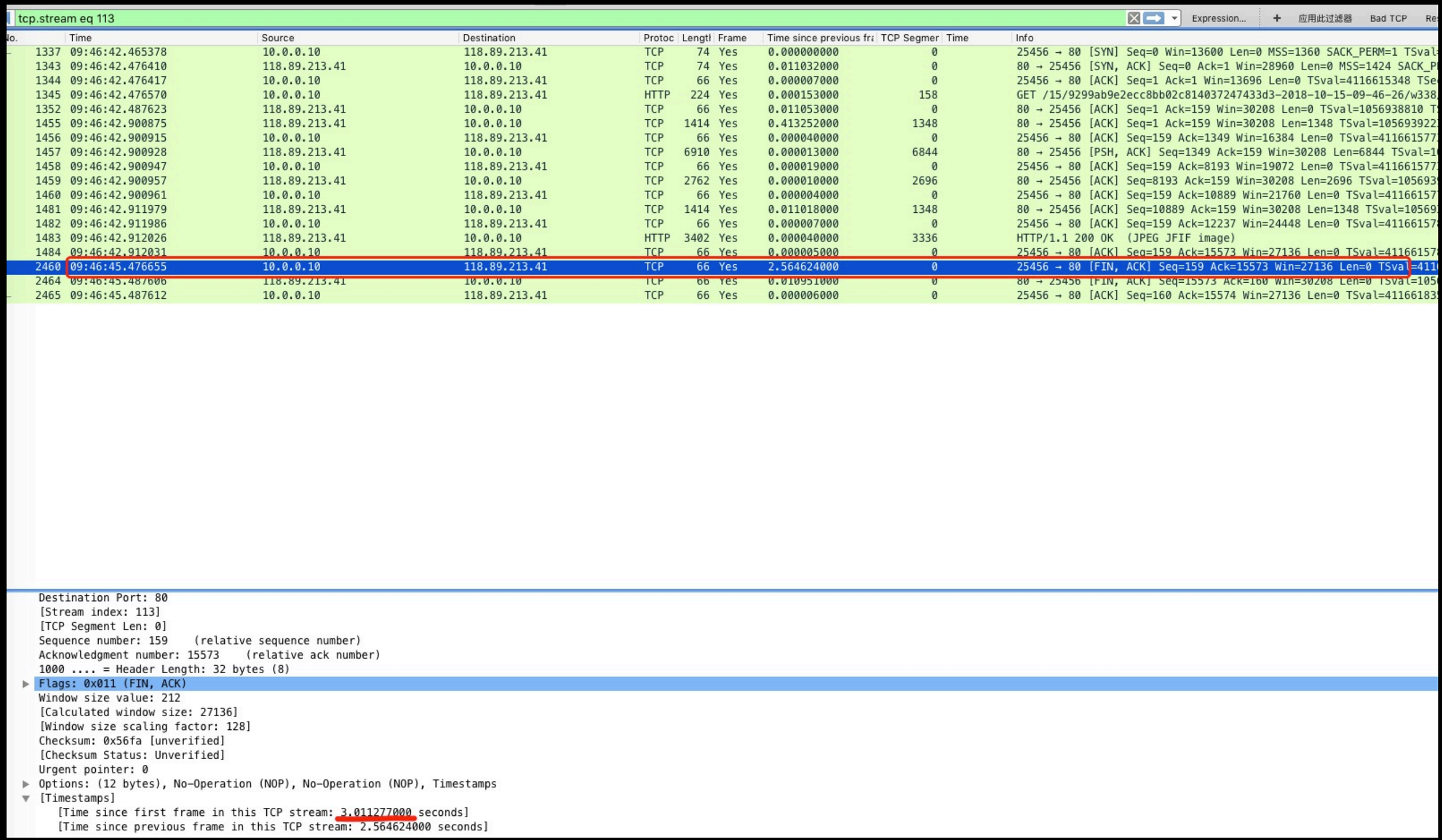
- 瀑布模型 https://www.webpagetest.org
- 域名分区 队首阻塞问题
- HTTP 管道 客户端 + 服务端支持
- HTTP 2.0

GO Http Client

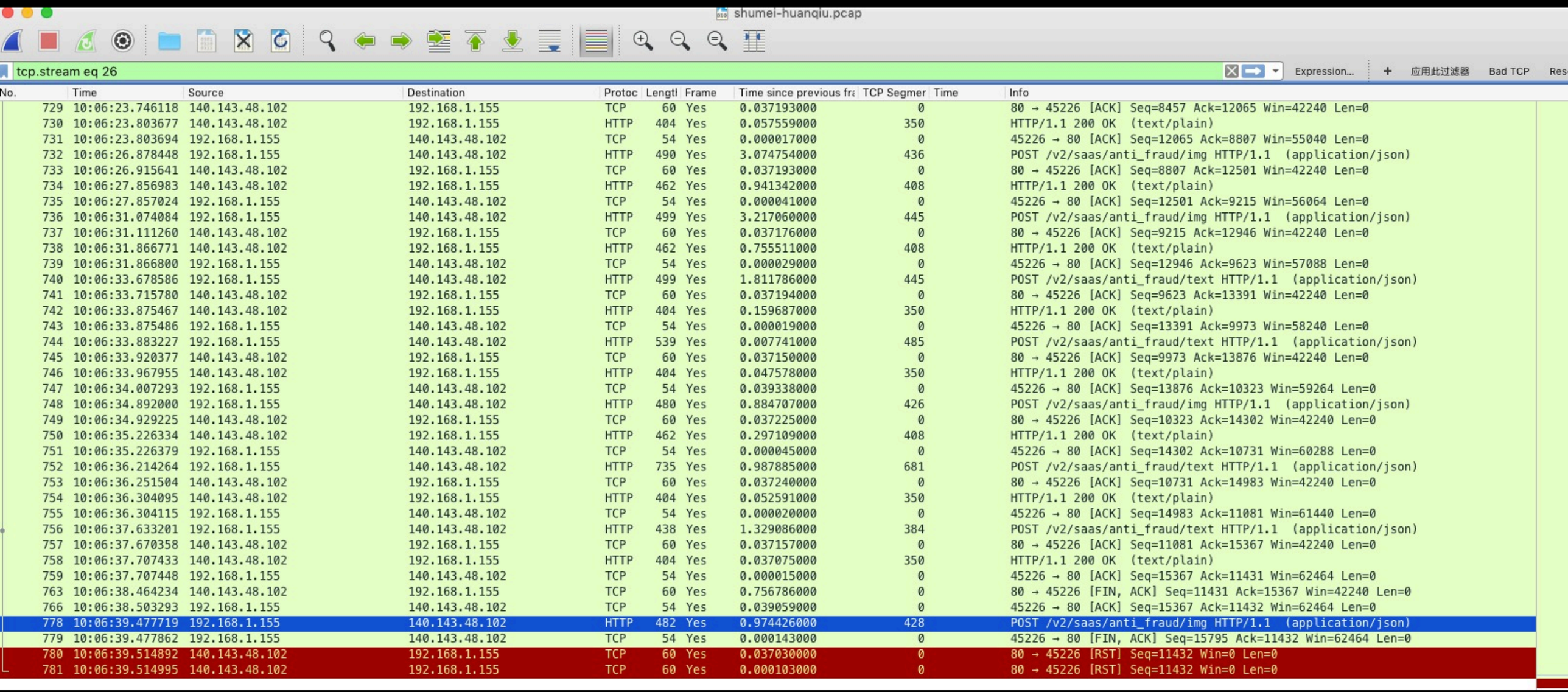
- goroutine safe
- 内置重用链接

```go
func NewHttpClient(connTimeout time.Duration, readTimeout time.Duration) HttpClient {
    client := http.Client{
        Transport: &http.Transport{
            Dial: func(netw, addr string) (net.Conn, error) {
                c, err := net.DialTimeout("tcp4", addr, connTimeout)
                if err != nil {
                    return nil, err
                }
                c.SetDeadline(time.Now().Add(readTimeout))
                return c, nil
            },
        },
    }
    return HttpClient{Client: client}
}
func NewHttpClient2(connTimeout time.Duration) HttpClient {
    client := http.Client{
        Transport: &http.Transport{
            DialContext: func(ctx context.Context, netw, addr string) (net.Conn, error) {
                c, err := net.DialTimeout("tcp4", addr, connTimeout)
                if err != nil {
                    return nil, err
                }
                return c, nil
            },
            DisableCompression:  true,
            MaxIdleConnsPerHost: 100,
            IdleConnTimeout:     30 * time.Second,
        },
        Timeout: 60 * time.Second,
    }
    return HttpClient{Client: client}
}
```

GO Http Client

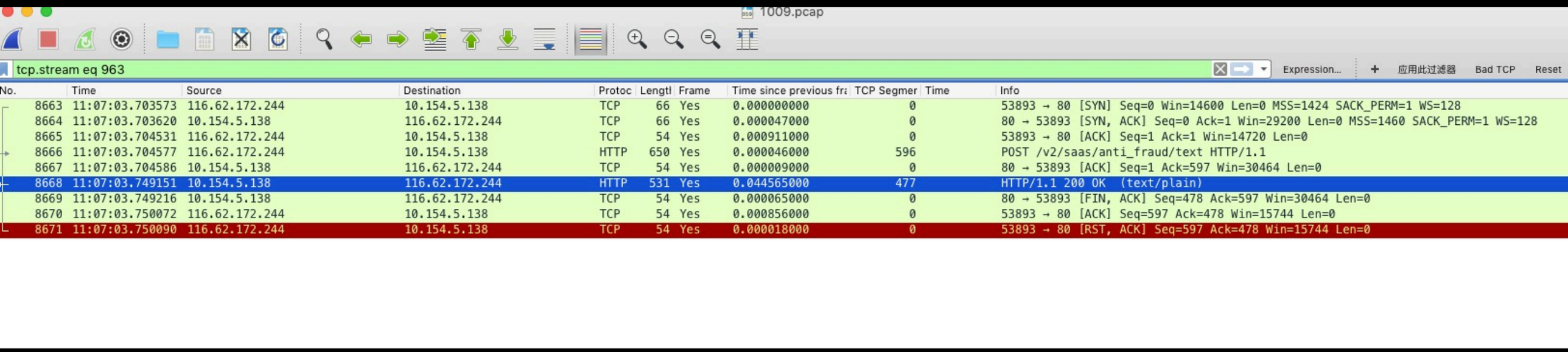


WireShark 分析

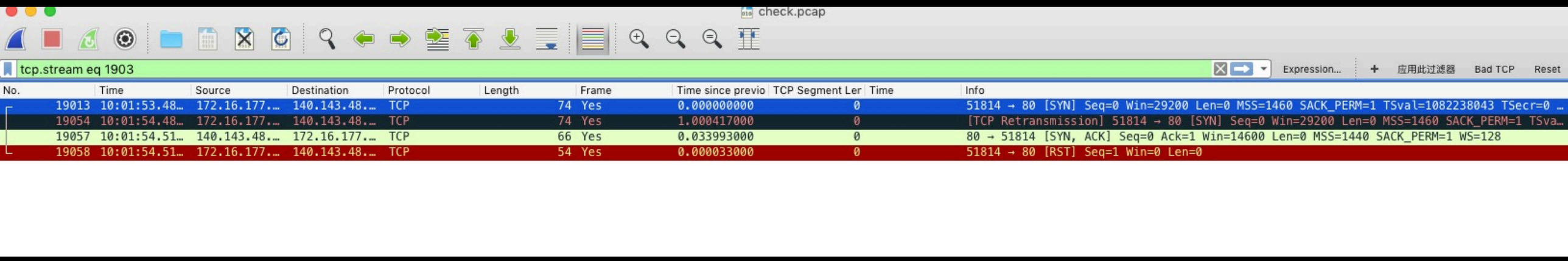


No.	Time	Source	Destination	Protoc	Lengtl	Frame	Time since previous fra	TCP Segmer	Time	Info
729	10:06:23.746118	140.143.48.102	192.168.1.155	TCP	60	Yes	0.037193000	0		80 → 45226 [ACK] Seq=8457 Ack=12065 Win=42240 Len=0
730	10:06:23.803677	140.143.48.102	192.168.1.155	HTTP	404	Yes	0.057559000	350		HTTP/1.1 200 OK (text/plain)
731	10:06:23.803694	192.168.1.155	140.143.48.102	TCP	54	Yes	0.000017000	0		45226 → 80 [ACK] Seq=12065 Ack=8807 Win=55040 Len=0
732	10:06:26.878448	192.168.1.155	140.143.48.102	HTTP	490	Yes	3.074754000	436		POST /v2/saas/anti_fraud/img HTTP/1.1 (application/json)
733	10:06:26.915641	140.143.48.102	192.168.1.155	TCP	60	Yes	0.037193000	0		80 → 45226 [ACK] Seq=8807 Ack=12501 Win=42240 Len=0
734	10:06:27.856983	140.143.48.102	192.168.1.155	HTTP	462	Yes	0.941342000	408		HTTP/1.1 200 OK (text/plain)
735	10:06:27.857024	192.168.1.155	140.143.48.102	TCP	54	Yes	0.000041000	0		45226 → 80 [ACK] Seq=12501 Ack=9215 Win=56064 Len=0
736	10:06:31.074084	192.168.1.155	140.143.48.102	HTTP	499	Yes	3.217060000	445		POST /v2/saas/anti_fraud/img HTTP/1.1 (application/json)
737	10:06:31.111260	140.143.48.102	192.168.1.155	TCP	60	Yes	0.037176000	0		80 → 45226 [ACK] Seq=9215 Ack=12946 Win=42240 Len=0
738	10:06:31.866771	140.143.48.102	192.168.1.155	HTTP	462	Yes	0.755511000	408		HTTP/1.1 200 OK (text/plain)
739	10:06:31.866800	192.168.1.155	140.143.48.102	TCP	54	Yes	0.000029000	0		45226 → 80 [ACK] Seq=12946 Ack=9623 Win=57088 Len=0
740	10:06:33.678586	192.168.1.155	140.143.48.102	HTTP	499	Yes	1.811786000	445		POST /v2/saas/anti_fraud/text HTTP/1.1 (application/json)
741	10:06:33.715780	140.143.48.102	192.168.1.155	TCP	60	Yes	0.037194000	0		80 → 45226 [ACK] Seq=9623 Ack=13391 Win=42240 Len=0
742	10:06:33.875467	140.143.48.102	192.168.1.155	HTTP	404	Yes	0.159687000	350		HTTP/1.1 200 OK (text/plain)
743	10:06:33.875486	192.168.1.155	140.143.48.102	TCP	54	Yes	0.000019000	0		45226 → 80 [ACK] Seq=13391 Ack=9973 Win=58240 Len=0
744	10:06:33.883227	192.168.1.155	140.143.48.102	HTTP	539	Yes	0.007741000	485		POST /v2/saas/anti_fraud/text HTTP/1.1 (application/json)
745	10:06:33.920377	140.143.48.102	192.168.1.155	TCP	60	Yes	0.037150000	0		80 → 45226 [ACK] Seq=9973 Ack=13876 Win=42240 Len=0
746	10:06:33.967955	140.143.48.102	192.168.1.155	HTTP	404	Yes	0.047578000	350		HTTP/1.1 200 OK (text/plain)
747	10:06:34.007293	192.168.1.155	140.143.48.102	TCP	54	Yes	0.039338000	0		45226 → 80 [ACK] Seq=13876 Ack=10323 Win=59264 Len=0
748	10:06:34.892000	192.168.1.155	140.143.48.102	HTTP	480	Yes	0.884707000	426		POST /v2/saas/anti_fraud/img HTTP/1.1 (application/json)
749	10:06:34.929225	140.143.48.102	192.168.1.155	TCP	60	Yes	0.037225000	0		80 → 45226 [ACK] Seq=10323 Ack=14302 Win=42240 Len=0
750	10:06:35.226334	140.143.48.102	192.168.1.155	HTTP	462	Yes	0.297109000	408		HTTP/1.1 200 OK (text/plain)
751	10:06:35.226379	192.168.1.155	140.143.48.102	TCP	54	Yes	0.000045000	0		45226 → 80 [ACK] Seq=14302 Ack=10731 Win=60288 Len=0
752	10:06:36.214264	192.168.1.155	140.143.48.102	HTTP	735	Yes	0.987885000	681		POST /v2/saas/anti_fraud/text HTTP/1.1 (application/json)
753	10:06:36.251504	140.143.48.102	192.168.1.155	TCP	60	Yes	0.037240000	0		80 → 45226 [ACK] Seq=10731 Ack=14983 Win=42240 Len=0
754	10:06:36.304095	140.143.48.102	192.168.1.155	HTTP	404	Yes	0.052591000	350		HTTP/1.1 200 OK (text/plain)
755	10:06:36.304115	192.168.1.155	140.143.48.102	TCP	54	Yes	0.000020000	0		45226 → 80 [ACK] Seq=14983 Ack=11081 Win=61440 Len=0
756	10:06:37.633201	192.168.1.155	140.143.48.102	HTTP	438	Yes	1.329086000	384		POST /v2/saas/anti_fraud/text HTTP/1.1 (application/json)
757	10:06:37.670358	140.143.48.102	192.168.1.155	TCP	60	Yes	0.037157000	0		80 → 45226 [ACK] Seq=11081 Ack=15367 Win=42240 Len=0
758	10:06:37.707433	140.143.48.102	192.168.1.155	HTTP	404	Yes	0.037075000	350		HTTP/1.1 200 OK (text/plain)
759	10:06:37.707448	192.168.1.155	140.143.48.102	TCP	54	Yes	0.000015000	0		45226 → 80 [ACK] Seq=15367 Ack=11431 Win=62464 Len=0
763	10:06:38.464234	140.143.48.102	192.168.1.155	TCP	60	Yes	0.756786000	0		80 → 45226 [FIN, ACK] Seq=11431 Ack=15367 Win=42240 Len=0
766	10:06:38.503293	192.168.1.155	140.143.48.102	TCP	54	Yes	0.039059000	0		45226 → 80 [ACK] Seq=15367 Ack=11432 Win=62464 Len=0
778	10:06:39.477719	192.168.1.155	140.143.48.102	HTTP	482	Yes	0.974426000	428		POST /v2/saas/anti_fraud/img HTTP/1.1 (application/json)
779	10:06:39.477862	192.168.1.155	140.143.48.102	TCP	54	Yes	0.000143000	0		45226 → 80 [FIN, ACK] Seq=15795 Ack=11432 Win=62464 Len=0
780	10:06:39.514892	140.143.48.102	192.168.1.155	TCP	60	Yes	0.037030000	0		80 → 45226 [RST] Seq=11432 Win=0 Len=0
781	10:06:39.514995	140.143.48.102	192.168.1.155	TCP	60	Yes	0.000103000	0		80 → 45226 [RST] Seq=11432 Win=0 Len=0

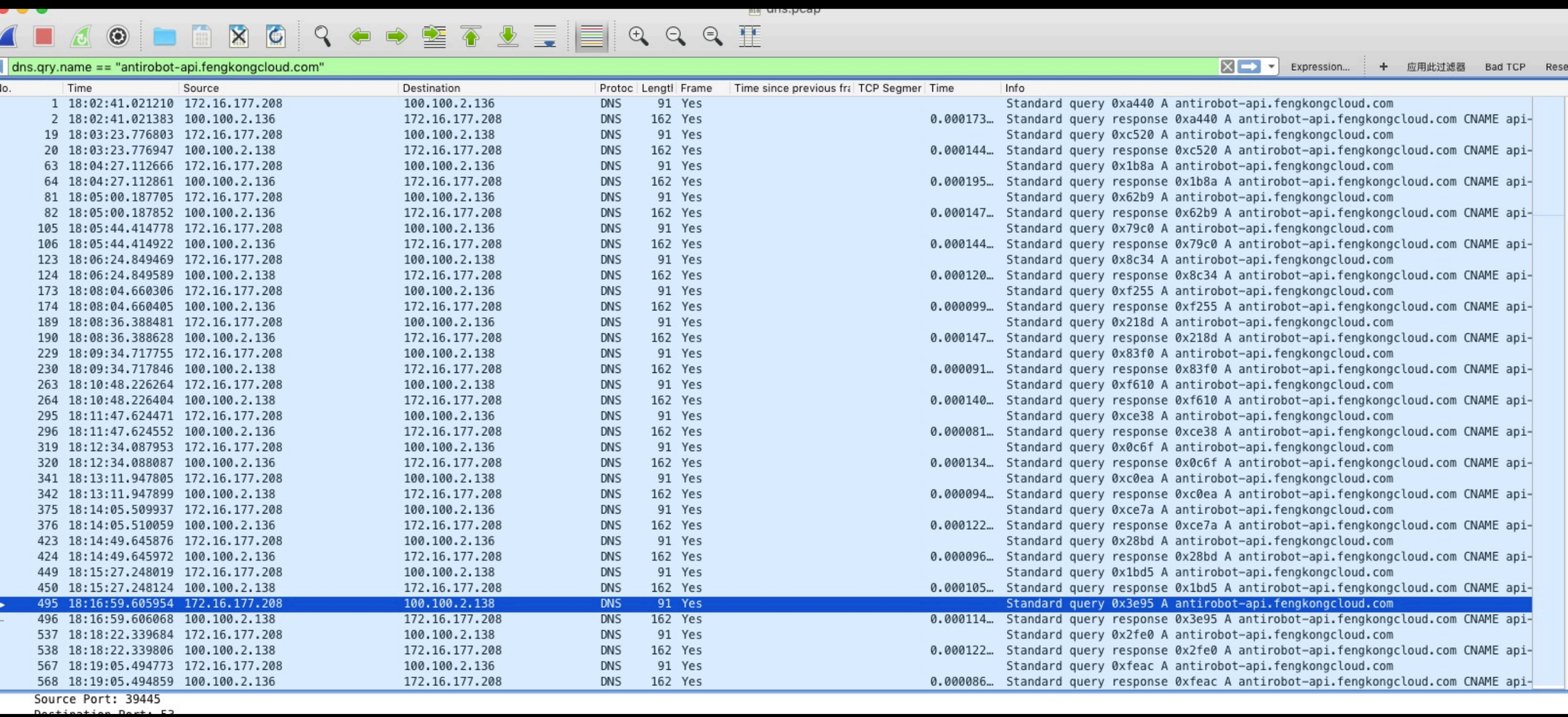
WireShark 分析



WireShark 分析



WireShark 分析



No.	Time	Source	Destination	Protoc	Lengtl	Frame	Time since previous fra	TCP Segmer	Time	Info
1	18:02:41.021210	172.16.177.208	100.100.2.136	DNS	91	Yes				Standard query 0xa440 A antirobot-api.fengkongcloud.com
2	18:02:41.021383	100.100.2.136	172.16.177.208	DNS	162	Yes			0.000173…	Standard query response 0xa440 A antirobot-api.fengkongcloud.com CNAME api-
19	18:03:23.776803	172.16.177.208	100.100.2.138	DNS	91	Yes				Standard query 0xc520 A antirobot-api.fengkongcloud.com
20	18:03:23.776947	100.100.2.138	172.16.177.208	DNS	162	Yes			0.000144…	Standard query response 0xc520 A antirobot-api.fengkongcloud.com CNAME api-
63	18:04:27.112666	172.16.177.208	100.100.2.136	DNS	91	Yes				Standard query 0x1b8a A antirobot-api.fengkongcloud.com
64	18:04:27.112861	100.100.2.136	172.16.177.208	DNS	162	Yes			0.000195…	Standard query response 0x1b8a A antirobot-api.fengkongcloud.com CNAME api-
81	18:05:00.187705	172.16.177.208	100.100.2.136	DNS	91	Yes				Standard query 0x62b9 A antirobot-api.fengkongcloud.com
82	18:05:00.187852	100.100.2.136	172.16.177.208	DNS	162	Yes			0.000147…	Standard query response 0x62b9 A antirobot-api.fengkongcloud.com CNAME api-
105	18:05:44.414778	172.16.177.208	100.100.2.136	DNS	91	Yes				Standard query 0x79c0 A antirobot-api.fengkongcloud.com
106	18:05:44.414922	100.100.2.136	172.16.177.208	DNS	162	Yes			0.000144…	Standard query response 0x79c0 A antirobot-api.fengkongcloud.com CNAME api-
123	18:06:24.849469	172.16.177.208	100.100.2.138	DNS	91	Yes				Standard query 0x8c34 A antirobot-api.fengkongcloud.com
124	18:06:24.849589	100.100.2.138	172.16.177.208	DNS	162	Yes			0.000120…	Standard query response 0x8c34 A antirobot-api.fengkongcloud.com CNAME api-
173	18:08:04.660306	172.16.177.208	100.100.2.136	DNS	91	Yes				Standard query 0xf255 A antirobot-api.fengkongcloud.com
174	18:08:04.660405	100.100.2.136	172.16.177.208	DNS	162	Yes			0.000099…	Standard query response 0xf255 A antirobot-api.fengkongcloud.com CNAME api-
189	18:08:36.388481	172.16.177.208	100.100.2.136	DNS	91	Yes				Standard query 0x218d A antirobot-api.fengkongcloud.com
190	18:08:36.388628	100.100.2.136	172.16.177.208	DNS	162	Yes			0.000147…	Standard query response 0x218d A antirobot-api.fengkongcloud.com CNAME api-
229	18:09:34.717755	172.16.177.208	100.100.2.138	DNS	91	Yes				Standard query 0x83f0 A antirobot-api.fengkongcloud.com
230	18:09:34.717846	100.100.2.138	172.16.177.208	DNS	162	Yes			0.000091…	Standard query response 0x83f0 A antirobot-api.fengkongcloud.com CNAME api-
263	18:10:48.226264	172.16.177.208	100.100.2.138	DNS	91	Yes				Standard query 0xf610 A antirobot-api.fengkongcloud.com
264	18:10:48.226404	100.100.2.138	172.16.177.208	DNS	162	Yes			0.000140…	Standard query response 0xf610 A antirobot-api.fengkongcloud.com CNAME api-
295	18:11:47.624471	172.16.177.208	100.100.2.136	DNS	91	Yes				Standard query 0xce38 A antirobot-api.fengkongcloud.com
296	18:11:47.624552	100.100.2.136	172.16.177.208	DNS	162	Yes			0.000081…	Standard query response 0xce38 A antirobot-api.fengkongcloud.com CNAME api-
319	18:12:34.087953	172.16.177.208	100.100.2.136	DNS	91	Yes				Standard query 0x0c6f A antirobot-api.fengkongcloud.com
320	18:12:34.088087	100.100.2.136	172.16.177.208	DNS	162	Yes			0.000134…	Standard query response 0x0c6f A antirobot-api.fengkongcloud.com CNAME api-
341	18:13:11.947805	172.16.177.208	100.100.2.138	DNS	91	Yes				Standard query 0xc0ea A antirobot-api.fengkongcloud.com
342	18:13:11.947899	100.100.2.138	172.16.177.208	DNS	162	Yes			0.000094…	Standard query response 0xc0ea A antirobot-api.fengkongcloud.com CNAME api-
375	18:14:05.509937	172.16.177.208	100.100.2.136	DNS	91	Yes				Standard query 0xce7a A antirobot-api.fengkongcloud.com
376	18:14:05.510059	100.100.2.136	172.16.177.208	DNS	162	Yes			0.000122…	Standard query response 0xce7a A antirobot-api.fengkongcloud.com CNAME api-
423	18:14:49.645876	172.16.177.208	100.100.2.136	DNS	91	Yes				Standard query 0x28bd A antirobot-api.fengkongcloud.com
424	18:14:49.645972	100.100.2.136	172.16.177.208	DNS	162	Yes			0.000096…	Standard query response 0x28bd A antirobot-api.fengkongcloud.com CNAME api-
449	18:15:27.248019	172.16.177.208	100.100.2.138	DNS	91	Yes				Standard query 0x1bd5 A antirobot-api.fengkongcloud.com
450	18:15:27.248124	100.100.2.138	172.16.177.208	DNS	162	Yes			0.000105…	Standard query response 0x1bd5 A antirobot-api.fengkongcloud.com CNAME api-
495	18:16:59.605954	172.16.177.208	100.100.2.138	DNS	91	Yes				Standard query 0x3e95 A antirobot-api.fengkongcloud.com
496	18:16:59.606068	100.100.2.138	172.16.177.208	DNS	162	Yes			0.000114…	Standard query response 0x3e95 A antirobot-api.fengkongcloud.com CNAME api-
537	18:18:22.339684	172.16.177.208	100.100.2.138	DNS	91	Yes				Standard query 0x2fe0 A antirobot-api.fengkongcloud.com
538	18:18:22.339806	100.100.2.138	172.16.177.208	DNS	162	Yes			0.000122…	Standard query response 0x2fe0 A antirobot-api.fengkongcloud.com CNAME api-
567	18:19:05.494773	172.16.177.208	100.100.2.136	DNS	91	Yes				Standard query 0xfeac A antirobot-api.fengkongcloud.com
568	18:19:05.494859	100.100.2.136	172.16.177.208	DNS	162	Yes			0.000086…	Standard query response 0xfeac A antirobot-api.fengkongcloud.com CNAME api-

Source Port: 39445

参考

- https://blog.cloudflare.com/the-story-of-one-latency-spike/
- http://confluence.ishumei.com:8090/pages/viewpage.action?pageId=11796635
- http://www.ece.virginia.edu/cheetah/documents/papers/TCPlinux.pdf
- https://access.redhat.com/sites/default/files/attachments/20150325_network_performance_tuning.pdf
- http://www.alloyteam.com/2012/03/web-cache-1-web-cache-overview/