2013 阿里技术沙龙
享技术·聚朋友

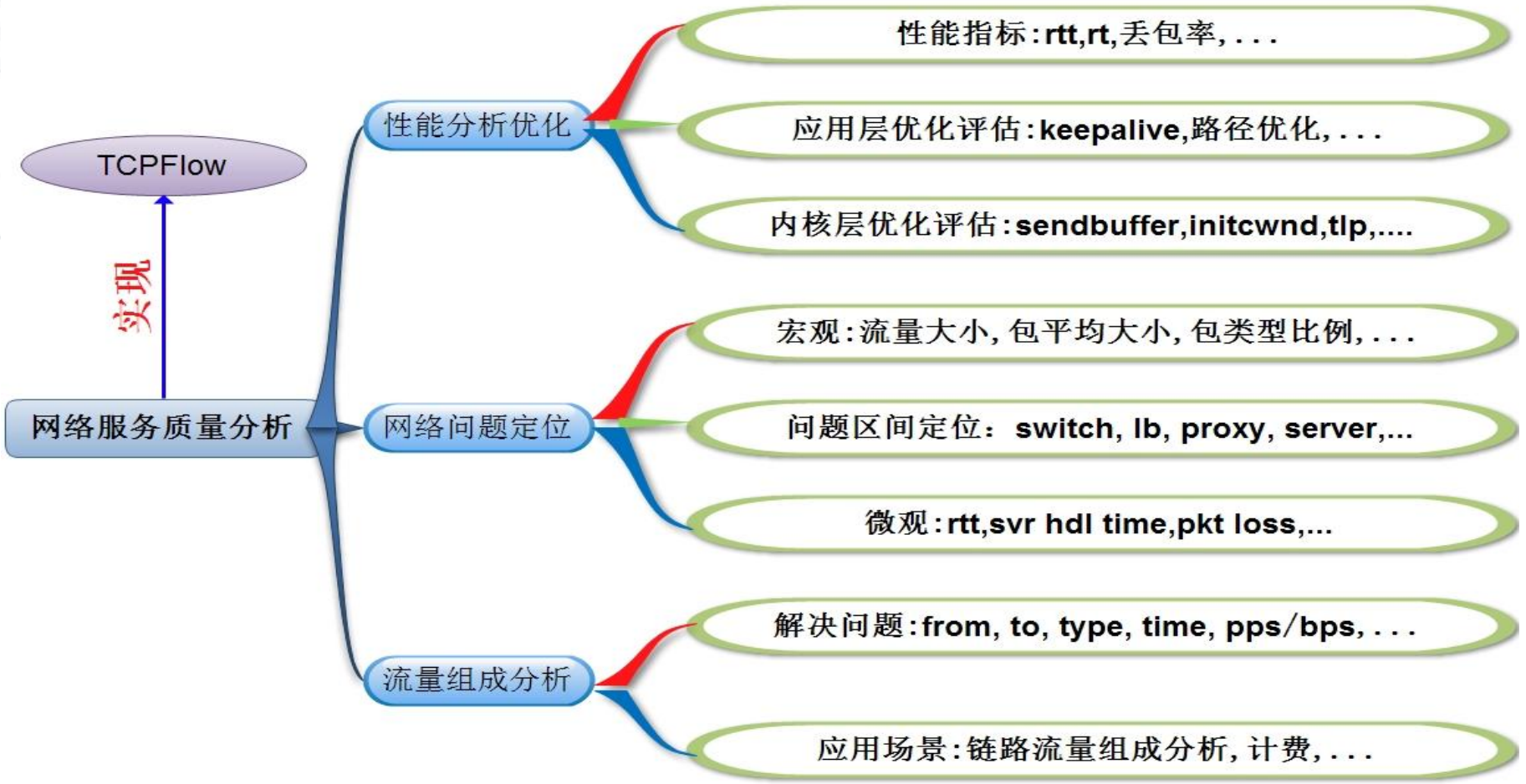
TCPFLOW业务网络服务质量分析

朱友志（花名：德泰）
阿里集团技术共享平台核心系统研发部网络组

目录

- 背景描述
- 功能描述
- 应用案例
- 设计要点
- 未来展望

背景描述：TCPFlow需求描述



TCPFlow需求之业务性能分析与优化

- 业务服务质量分析与优化为什么重要？
  - Amazon: 增加 100ms 延迟将导致收入下降1%；
  - Google: 400 ms 延迟将导致每用户搜索请求下降 0.59%；
  - Yahoo!: 400ms 延迟会导致流量下降 5-9%；
  - Bing: 2 秒的延迟将导致收入降低 4.3%/用户；
  - Shopzilla: 将页面载入时间从 7秒缩减到 2秒，转化率提升了 7-12%，页面请求增加 25%
  - ----IResearch《中国网络零售年志》

TCPFlow需求之业务性能分析与优化

- TCPFlow VS 其他业务性能获取方式

	探测类型	部署方	指标丰富性	覆盖面	可定制性	成本
NB/Gomez	主动	用户侧	★★★	★	★	★★★★
Aliprobe/UAQ	主动	用户侧	★★★	★	★★★	★★★★★
网页注入js	被动	用户侧	★★★	★★★★★	★★★	★★★
日志分析	被动	服务器侧	★	★★★★★	★★★	★★
TCPFlow流分析	被动	服务器侧	★★★	★★★★★	★★★★	★

TCPFlow需求之网络故障定位

- 需求描述：
  - 可用性变差（如http异常状态码增多：504， 408， …）
  - 服务质量参数变坏（rt变大，丢包率上升等）
- 与现有对比方法：
  - 第三方监控： 可定制性、覆盖范围
  - 日志： 信息量
  - tcpdump： 分析工作量、效率

TCPFlow需求之网络流量分析与可视化

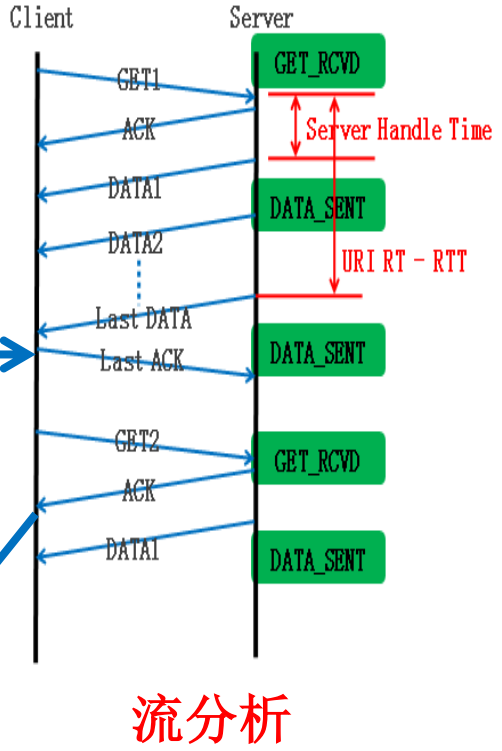
- 需求描述：
  - 流量是什么类型？从哪里来？到哪里去？什么时间？量多大？
- 粒度可控
  - 例子： Facebook关系图（细粒度、可视化的关系）

TCPFlow功能描述

- 对抓包结果进行流分析，输出日志

0.158186	110.[redacted]	218.[redacted]	TCP	9970 > http [SYN] Seq=2997769495 Win=5840 Len=0 MSS=1460 TSV=2538796562 T
0.158205	218.[redacted]	110.[redacted]	TCP	http > 9970 [SYN, ACK] Seq=2584289425 Ack=2997769496 Win=5840 Len=0 MSS=1
0.160384	110.[redacted]	218.[redacted]	TCP	9970 > http [ACK] Seq=2997769496 Ack=2584289426 Win=46 Len=0
0.160484	110.[redacted]	218.[redacted]	HTTP	GET /wangwang/wwupdate/new/7.20.21C/7.20.21C/0_chk.xml HTTP/1.1
0.160765	218.[redacted]	110.[redacted]	TCP	http > 9970 [ACK] Seq=2584289426 Ack=2997769602 Win=12 Len=0
0.161560	218.[redacted]	110.[redacted]	HTTP/XM	HTTP/1.1 200 OK [Packet size limited during capture]
0.164026	110.[redacted]	218.[redacted]	TCP	9970 > http [ACK] Seq=2997769602 Ack=2584292346 Win=69 Len=0
0.164040	218.[redacted]	110.[redacted]	HTTP	Continuation or non-HTTP traffic
0.166290	110.[redacted]	218.[redacted]	TCP	9970 > http [ACK] Seq=2997769602 Ack=2584292606 Win=115 Len=0
0.166349	110.[redacted]	218.[redacted]	TCP	9970 > http [FIN, ACK] Seq=2997769602 Ack=2584292606 Win=115 Len=0
0.166415	218.[redacted]	110.[redacted]	TCP	http > 9970 [FIN, ACK] Seq=2584292606 Ack=2997769603 Win=12 Len=0
0.168645	110.[redacted]	218.[redacted]	TCP	9970 > http [ACK] Seq=2997769603 Ack=2584292607 Win=115 Len=0

抓包结果



流分析

输出分析日志

```
HTTP    1354773338 937177 1354773339    373[redacted].36209      370[redacted].80   1      28
500     42040   3       GET     /[redacted]-200.png  200     29      0
10      0       65      21      1       28      1
```

TCPFlow功能描述 – 输出通用信息

- 通用信息

序号	名称	样例	备注
1	type	HTTP	日志类型
2	timestamp-sec(create)	1339584415	创建时间戳秒数（1970-01-01 以来秒数）
3	timestamp-usec(create)	259691	创建时间戳微秒数
4	timestamp-sec(output)	1339584419	打印时间戳秒数（1970-01-01 以来秒数）
5	sip.port	12345678912.54520	源 ip(整数)和 port
6	dip.port	987654321987.80	目的 ip(整数)和 port
7	packets_in	1	一个 get 请求时收到的数据包个数
8	packets_out	1	一个 get 请求时发送的数据包个数
9	bytes_in	640	收到的 http 请求字节数
10	bytes_out	640	发送的 http 回应字节数

TCPFlow功能描述 – TCP相关字段

TCP信息

11	Syn-flag_in	10	收到的 Syn 包个数（被监测 ip）
12	Syn-flag_out	10	发送的 Syn 包个数（被监测 ip）
13	Syn-Ack-flag_in	100	收到的 syn-ack 包个数（被监测 ip）
14	Syn-Ack-flag_out	100	发送的 syn-ack 包个数（被监测 ip）
15	Ack-flag_in	100	收到的 ack 包个数（被监测 ip）
16	Ack-flag_out	100	发送的 ack 包个数（被监测 ip）
17	Fin-flag_in	10	收到的 Fin 包个数（被监测 ip）
18	Fin-flag_out	10	发送的 Fin 包个数（被监测 ip）
19	Fin-ack-flag_in	10	收到的 Fin-ack 包个数（被监测 ip）
20	Fin-ack-flag_out	10	发送的 Fin-ack 包个数（被监测 ip）
21	Rst-flag_in	10	收到的 Reset 包个数（被监测 ip）
22	Rst-flag_out	10	发送的 Reset 包个数（被监测 ip）
23	Rst-ack flag in	10	收到的 rst-ack 包个数（被监测 ip）
24	Rst-ack flag out	10	发送的 rst-ack 包个数（被检测 ip）
25	connection_loss_pkt_num	2	建立连接阶段丢包数
26	data_loss_packet_num	10	数据传输阶段丢包数
27	close_loss_packet_num	2	关闭连接阶段丢包数
28	rtt	10	Round-trip-time(往返时间)
29	ct	15	建立连接时间(ms)
30	ttl	32	IP 包中的 ttl 字段
31	tcp offset	1	同一个 tcp 流的打印序号（从 1 开始）
32	http request num	1	同一个 tcp 流中的 http 请求和响应个数

TCPFlow功能描述 – HTTP相关信息

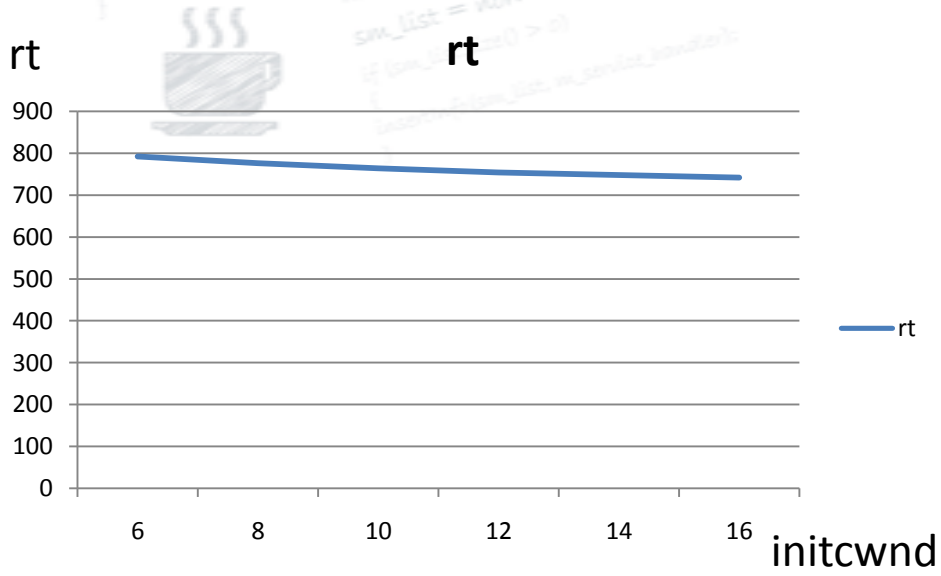
- HTTP信息

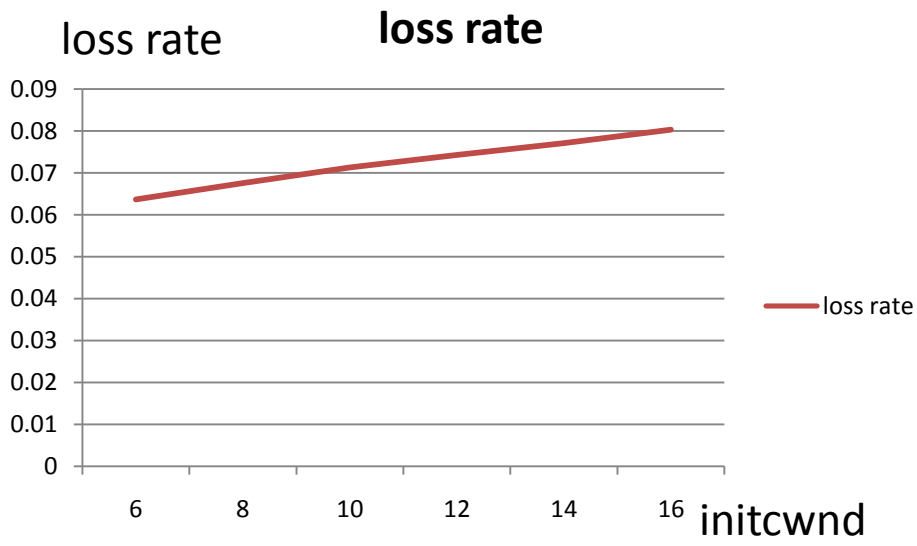
11	**http net body bytes**	**640**	http 七层响应体字节数
12	**http offest**	**1**	同一个 tcp 流中第几个 http
13	**http request**	**GET/POST**	HTTP 请求类型
14	**Host**	**img01.taobaocdn.com**	http 请求的 host 信息
15	**http URI**	**/1.jpg**	URI 信息
16	**http response**	**200/403/……**	HTTP 应答 code（当没有响应时返回 N/A）
17	**http total data packets**	**506**	HTTP 数据包的总数（含 GET 请求）
18	**http loss data packets**	**4**	HTTP 请求和响应丢失的数据包数(含 GET 请求)
19	**http out loss packets**	**2**	HTTP 响应的丢包数
20	**http init wnd**	**3**	HTTP 响应在收到第一个 ack 前发送的包数
21	**server handle time**	**4**	服务器收到 GET 请求到开始发送数据时间(ms)
22	**rt**	**5**	HTTP 请求的响应时间(ms)

TCPFlow应用案例 – 优化评估 - InitCWnd调整评估

- InitCWnd调整效果: InitCWnd合理值如何选择?

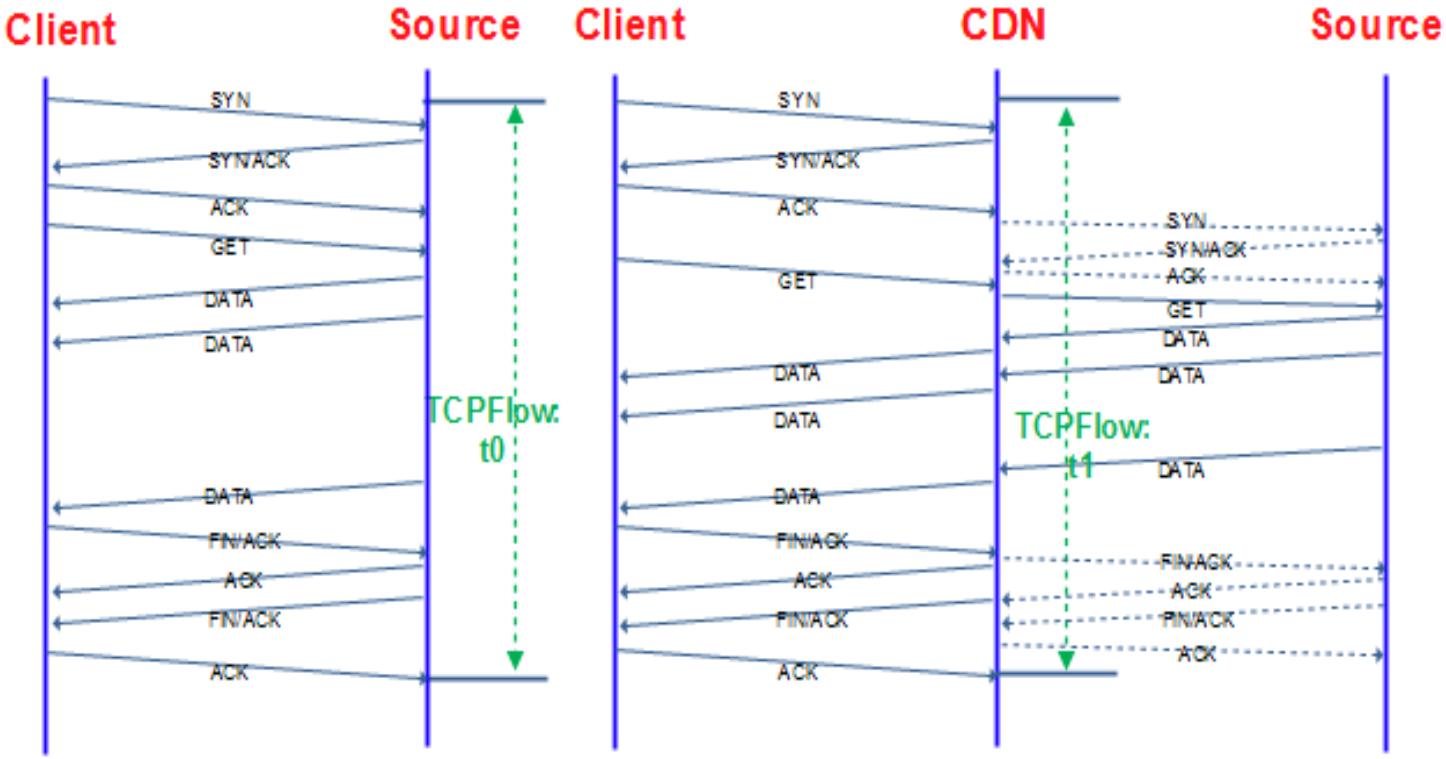
initcwnd	rt(ms)	loss rate
6	792	6.37%
8	776	6.76%
10	764	7.13%
12	754	7.43%
14	748	7.71%
16	742	8.03%





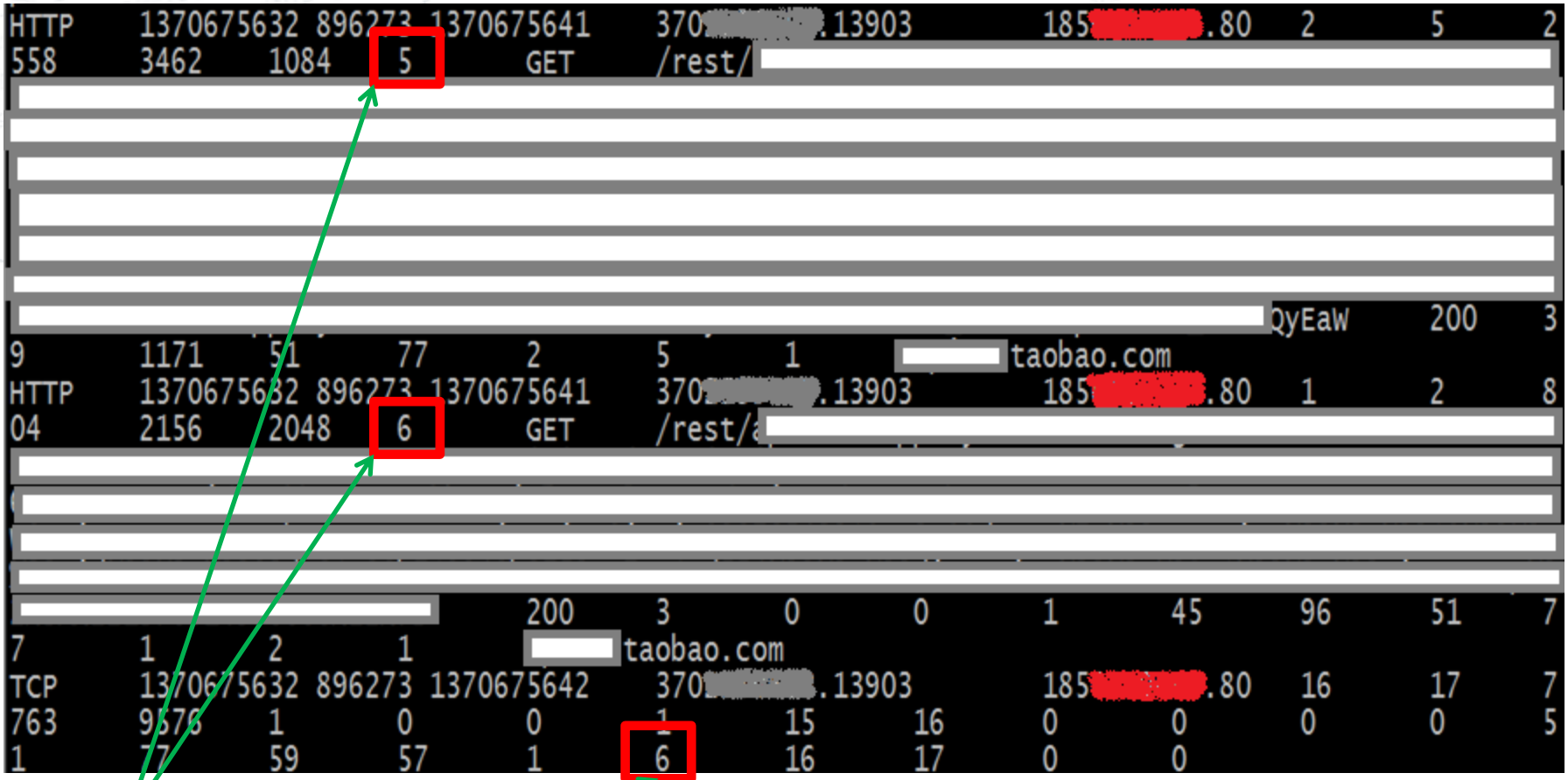
TCPFlow应用案例 – 优化评估 - 动态加速效果评估

- 利用TCPFlow测试动态内容加速效果(t1 VS t0)



TCPFlow应用案例 – 优化评估 - http keep-alive开启效果评估

- 长连接开启是否有利于改善业务rt？



请求序号　　同一个连接请求数

TCPFlow应用案例 - 网络问题定位 – 服务器处理时间过长

- 服务器处理时间过长导致rt大

HTTP 1370673260 778163 1370673310 190*******.1728 185*******.80 1 1
1257 265 211 1 GET
/sub?nick=**********
********** 200 2 0
0 **50000** **50053** 70 105 1 1 1 ****.taobao.com

服务器处理时间 rt

TCPFlow应用案例 – 网络问题定位 – rtt过大

- rtt过大导致rt过大

HTTP 1370675470 28730 1370675475 826******.2300 185*******.80 1 2
1235 1132 512 1 GET
/tbc?callback=********** 200 15 **3950** **2338**
3511 1 2 1 ****.taobao.com

rtt

rt

TCPFlow应用案例 - 网络问题定位 – 丢包和乱序

- http传输阶段丢包、乱序严重

HTTP 1370675628 483486 1370675641 192*******.15689 185*******.80 2
36 1582 50837 22825 1 GET /?spm=********** 200 **38**
19 19 10 36 **13459** 16 24 2 36 1 ****.taobao.com

数据包丢包乱序包数

rt

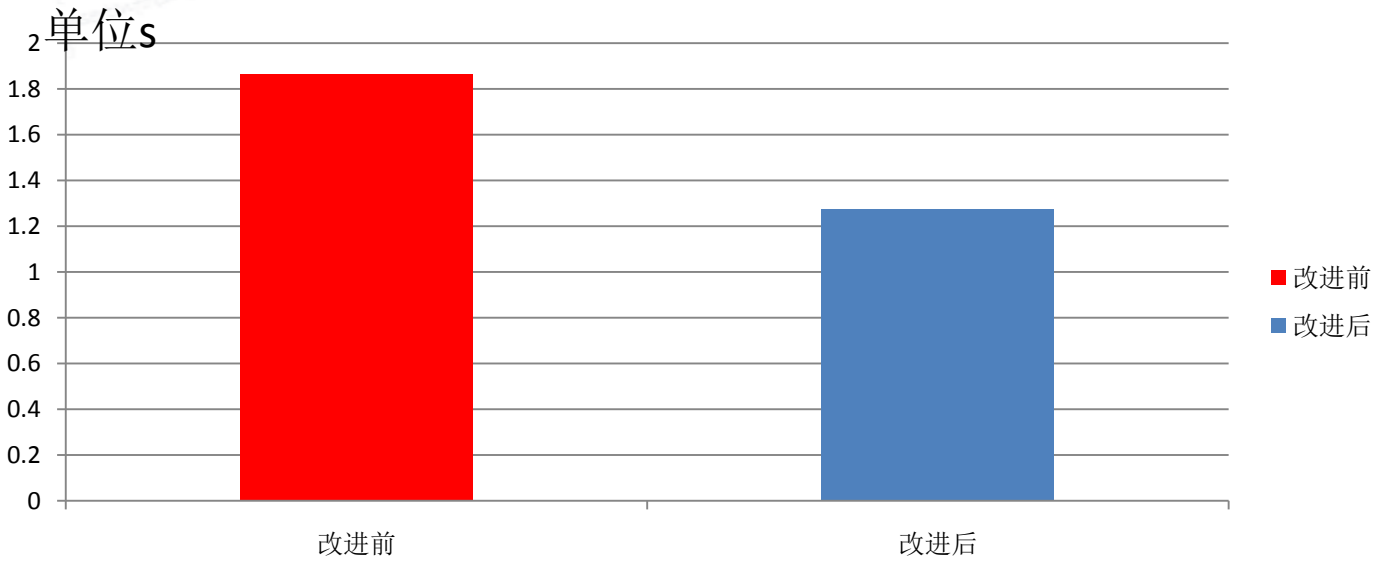
数据包总包数

TCPFlow应用案例 – 网络问题定位 – DNS调度错误

- 发现调度错误改进rt
  - 通过TCPFlow分析抓包结果发现City1 ISP 1节点有部分来自City2 ISP2 的访问(3402525012.55352<->200*******.80)
  - 改进调度策略，将City 2 ISP2 的用户调度到就近的ISP 2节点
- 对象rt改进效果

City2 ISP2

City 1 ISP 1



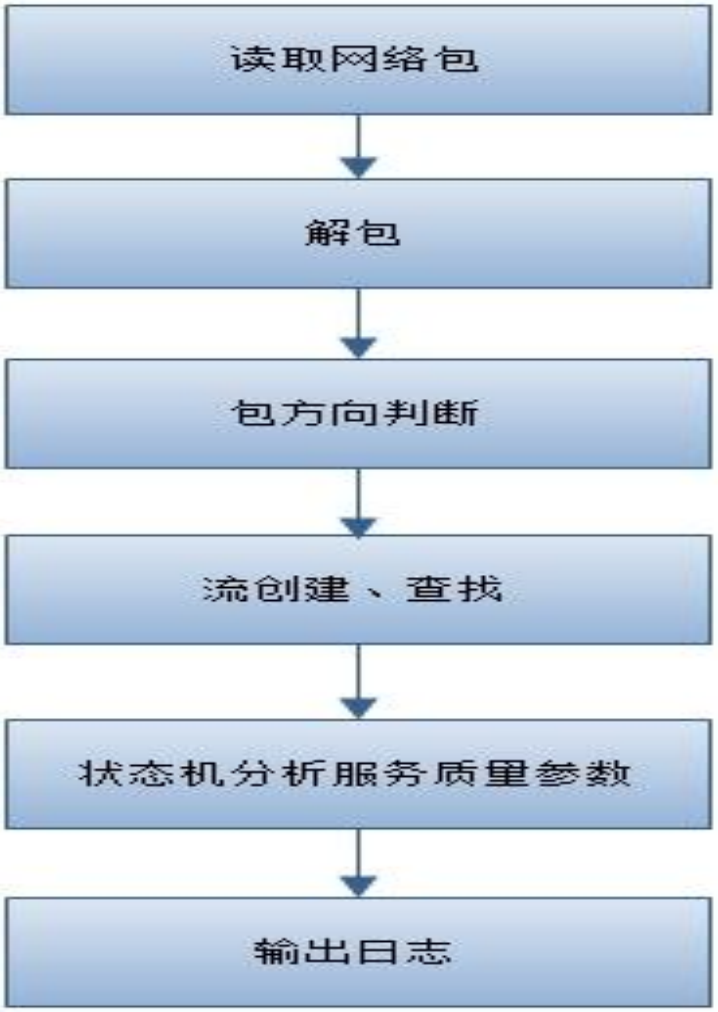
TCPFlow应用案例 - 流量组成分析

- 链路带宽利用率高
  - 场景描述：网络上某条链路带宽利用率一直处于非常高的水平，需要摸清流量组成。
  - 解决办法：对链路数据抓包，在利用tcpflow分析二元组组流量top 10
  - 结果：内部通信占用了大量的带宽！！

IP1	IP2	Bytes	Ratio
172.30.38.22	172.20.130.78	4516416	**0.074**
172.30.38.14	172.20.135.6	4226218	**0.069**
172.22.5.178	172.20.142.4	3159775	**0.052**
172.30.38.22	172.20.132.31	1843042	**0.030**
172.16.198.130	172.20.130.68	1472316	**0.024**
10.228.90.25	172.20.233.1	1293990	**0.021**
172.30.38.23	172.20.130.78	1259080	**0.021**
172.16.198.130	172.20.130.65	1209308	**0.020**

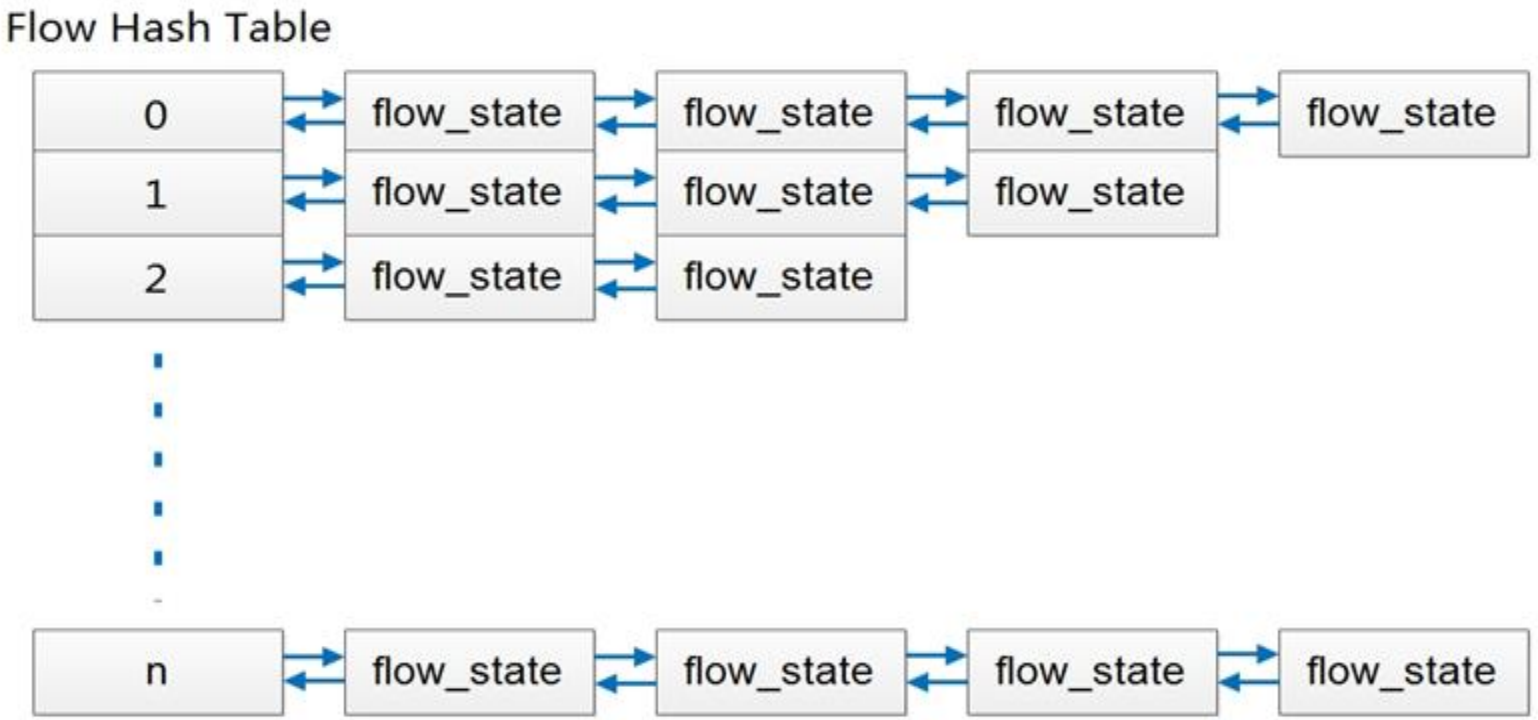
TCPFlow设计要点 – 整体流程

- 流分析整理流程



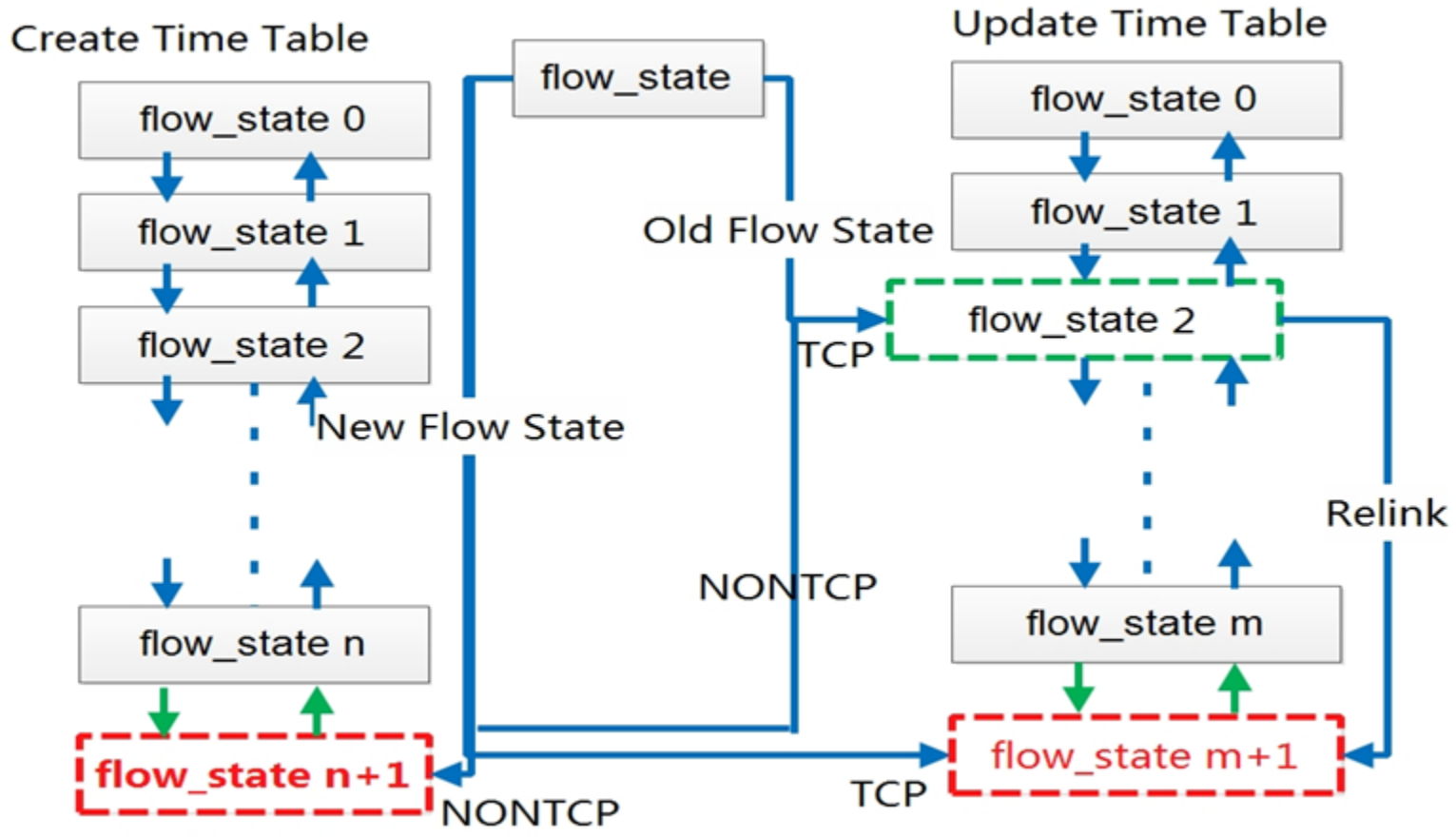
TCPFlow设计要点 – 流表示、存储与查找

- 通信五元组存储与查找
- (proto, sip, sport, dip, dport)

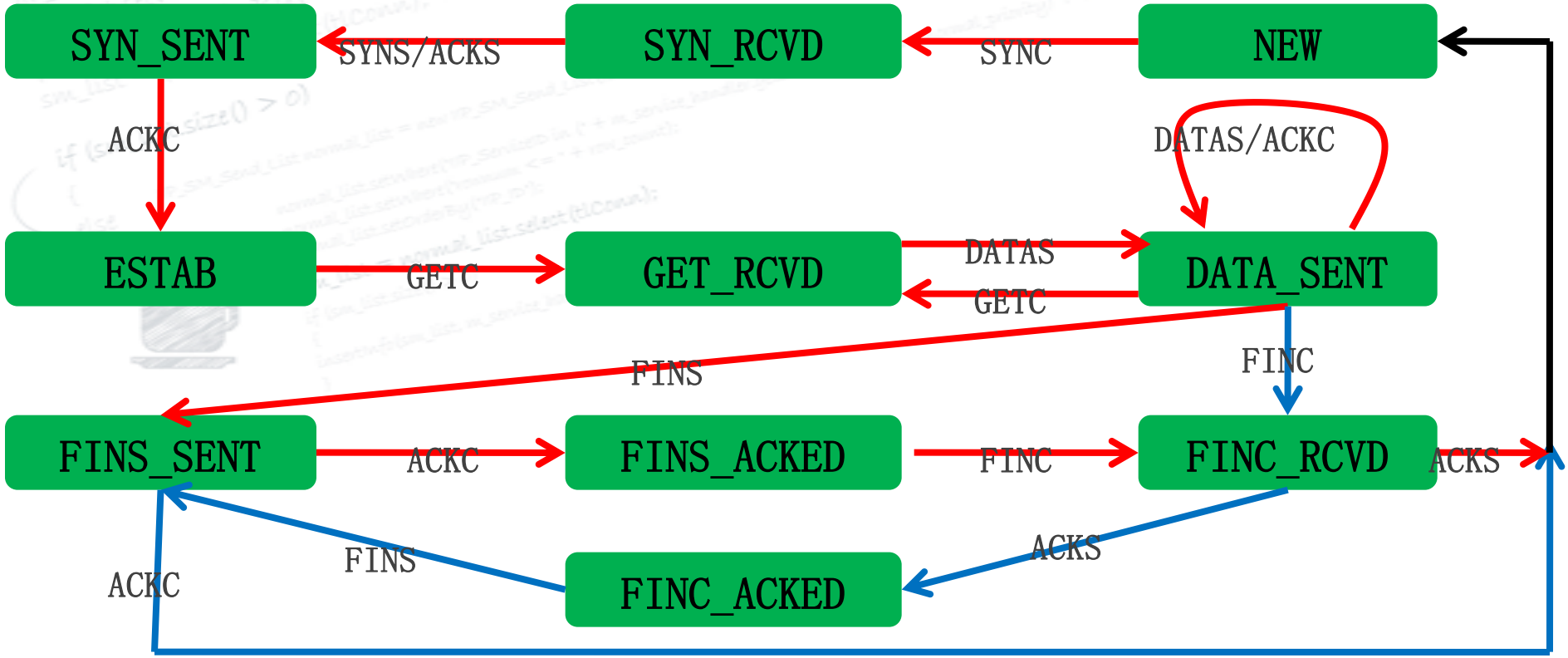


TCPFlow设计要点 – 超时处理

- 创建时间与更新时间



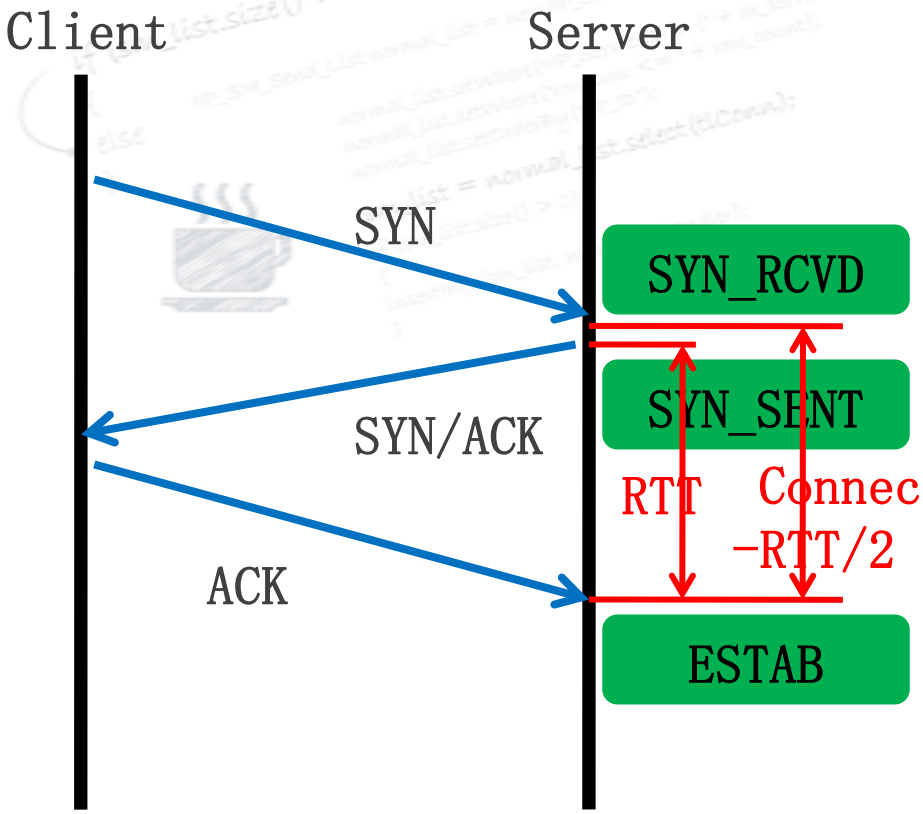
TCPFlow设计要点 – 服务质量参数计算 - 状态机设计



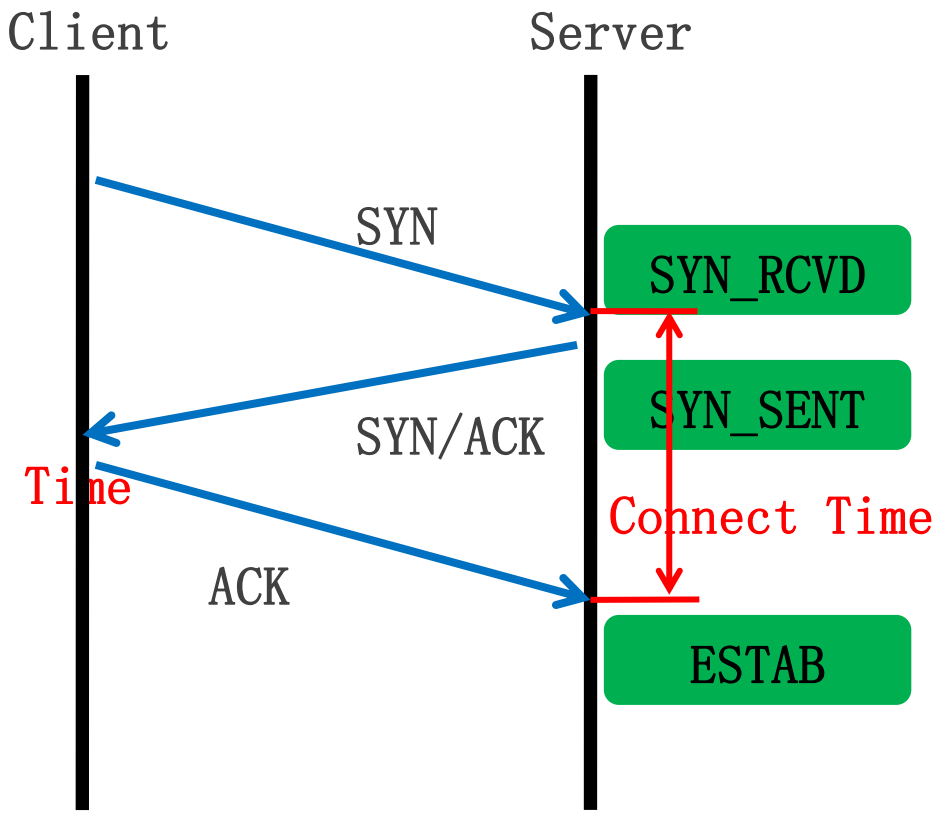
TCPFlow设计要点 – 服务质量参数计算 - 建立连接

- 建立连接时间定义

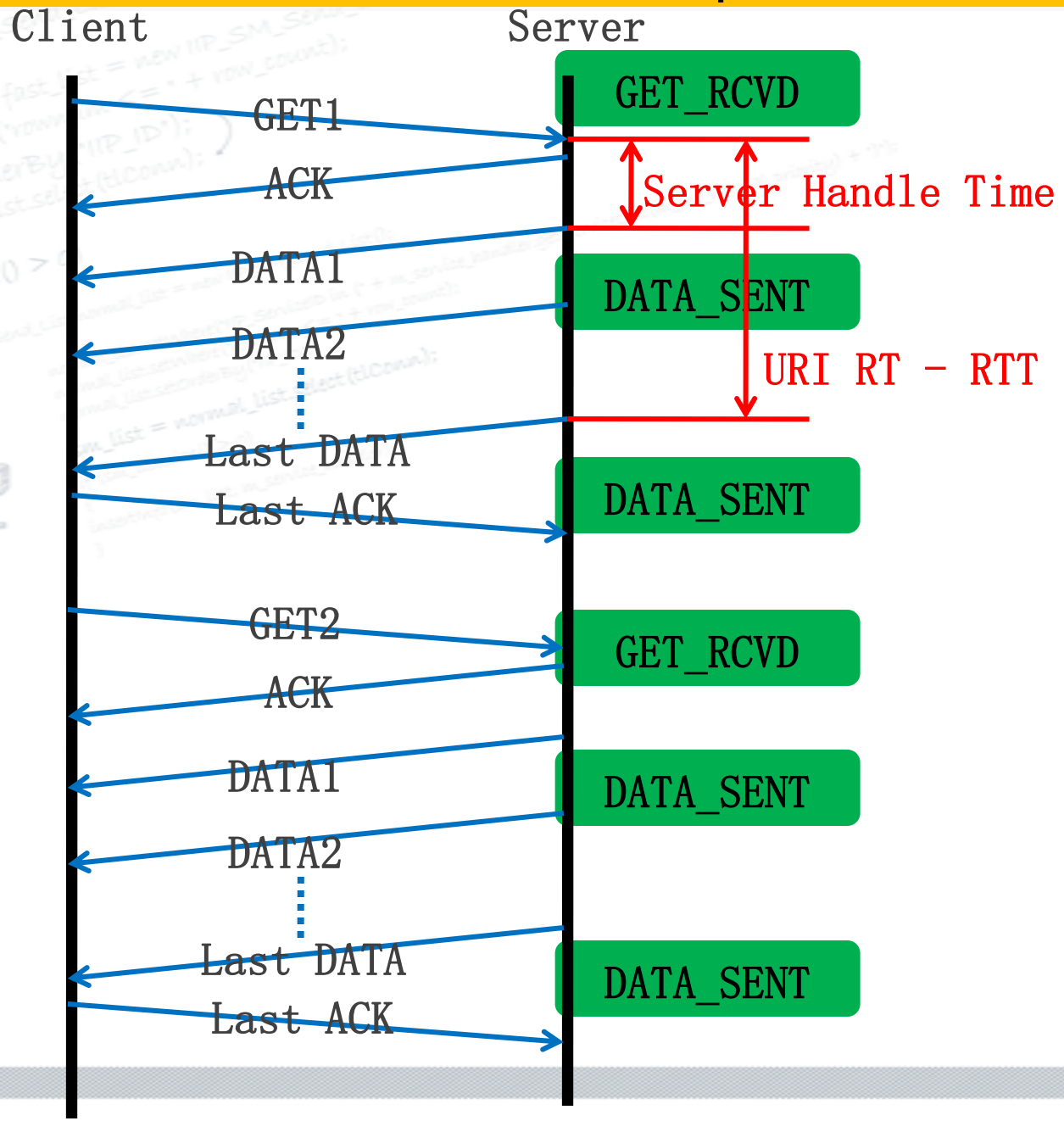
定义1



定义2

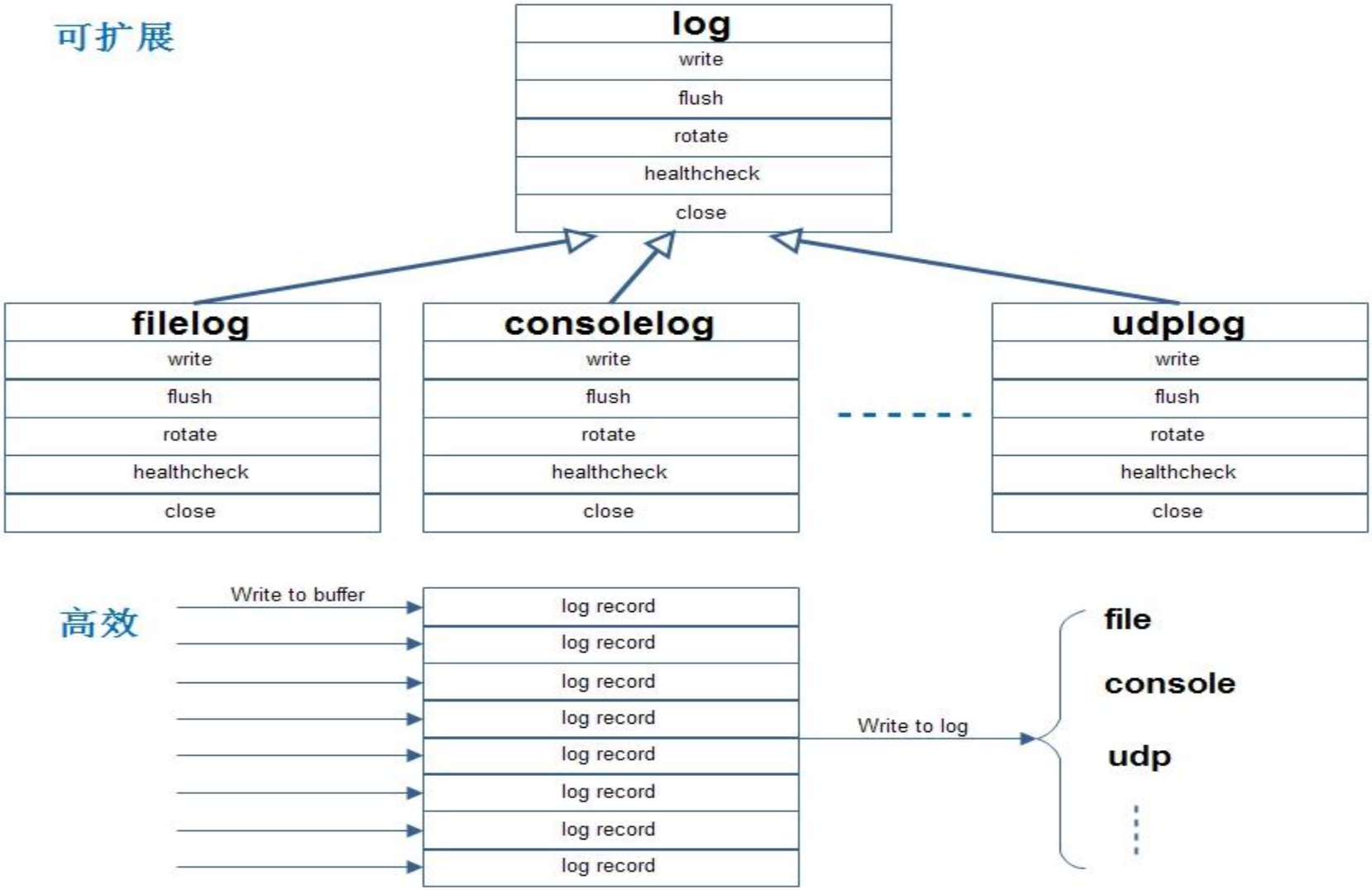


TCPFlow设计要点 – 服务质量参数 - http请求时间参数



TCPFlow设计要点 – 日志模块

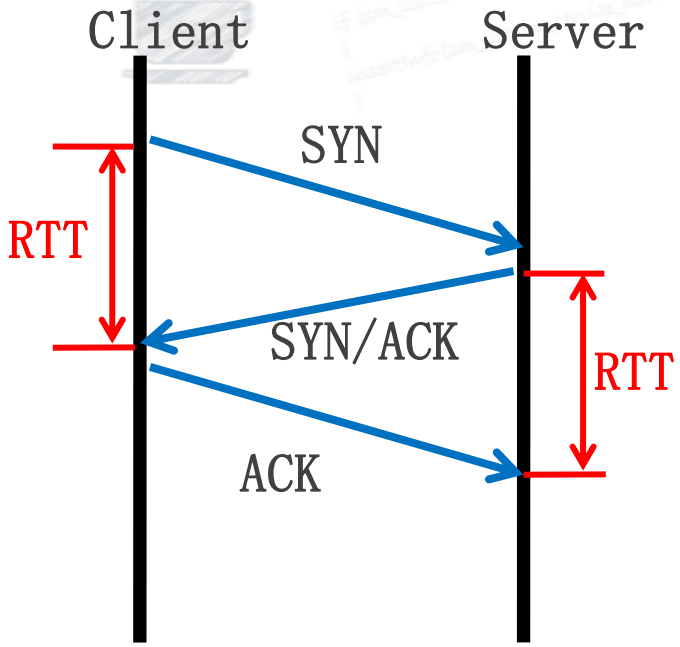
- 可扩展的日志模块



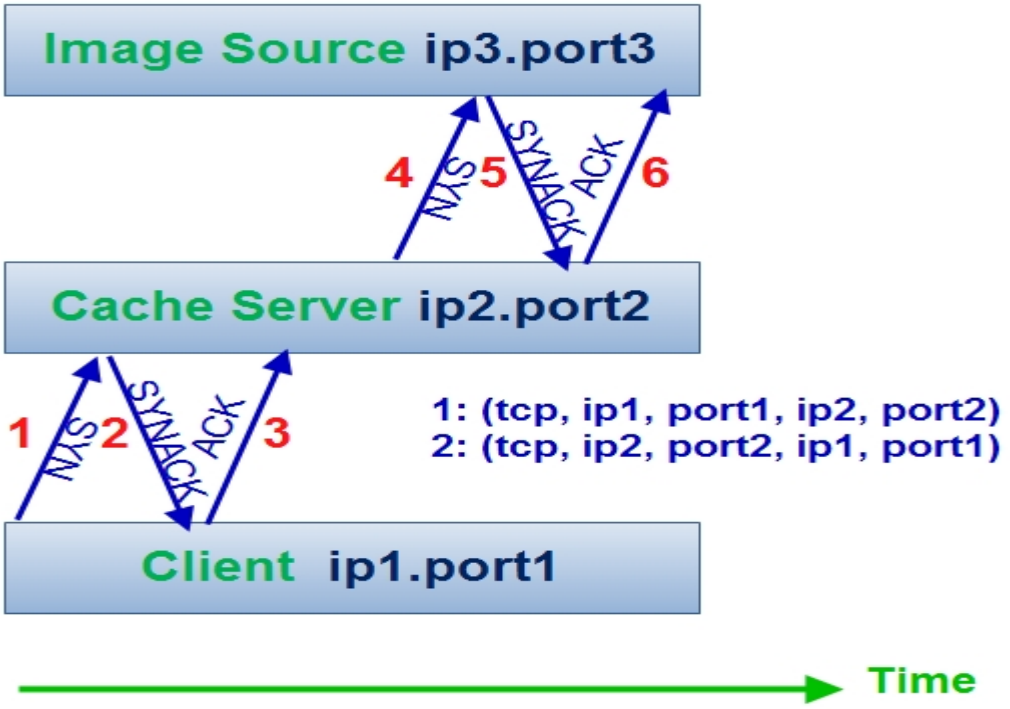
TCPFlow设计考虑 – 角色与包方向

- 部署在server端和client端对流分析有什么影响？
- 如何确定服务器端？
- 包方向为什么重要？
- 如何确定包的方向？

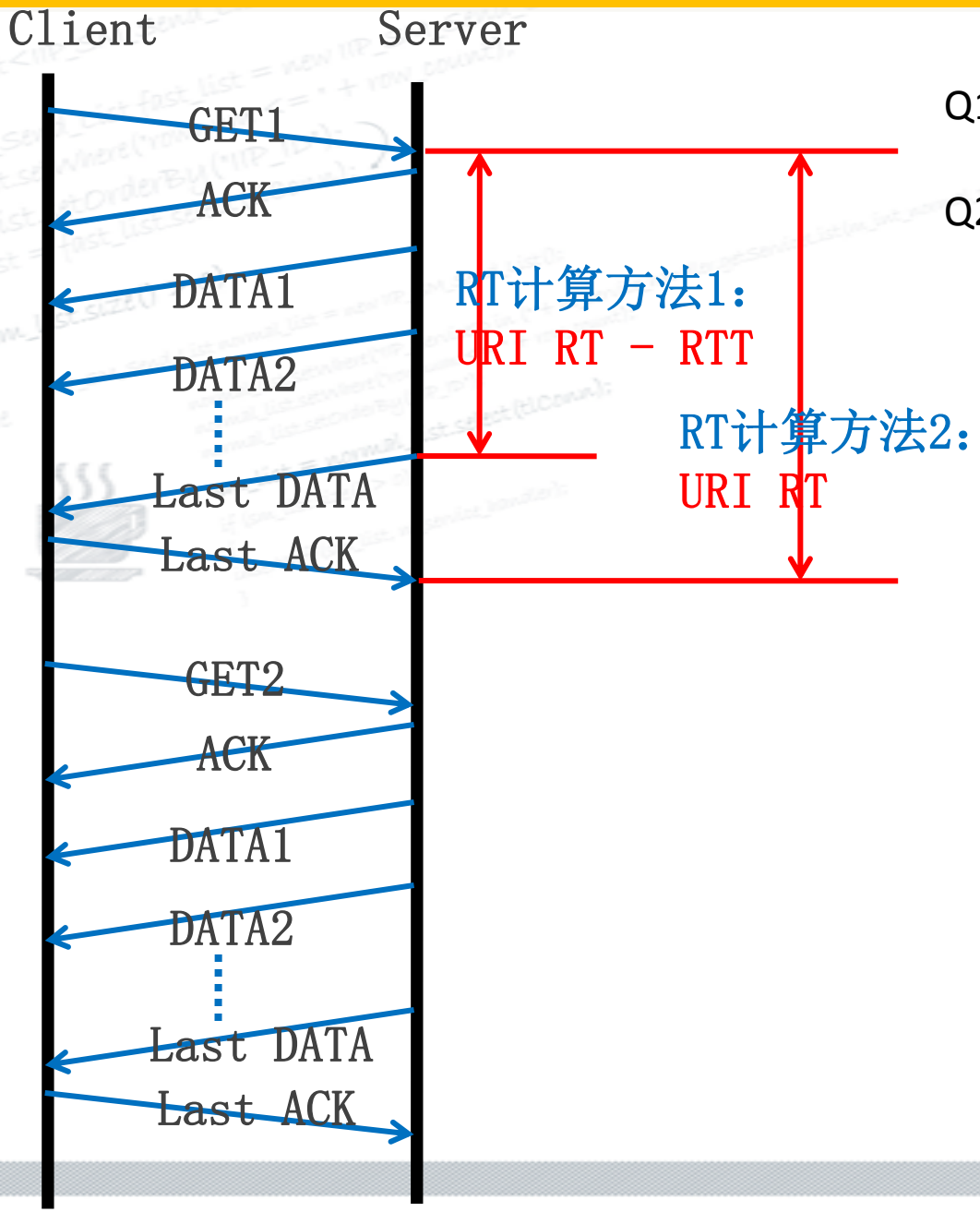
RTT BY Client: RTT BY Server



A Server act as both Client & Server



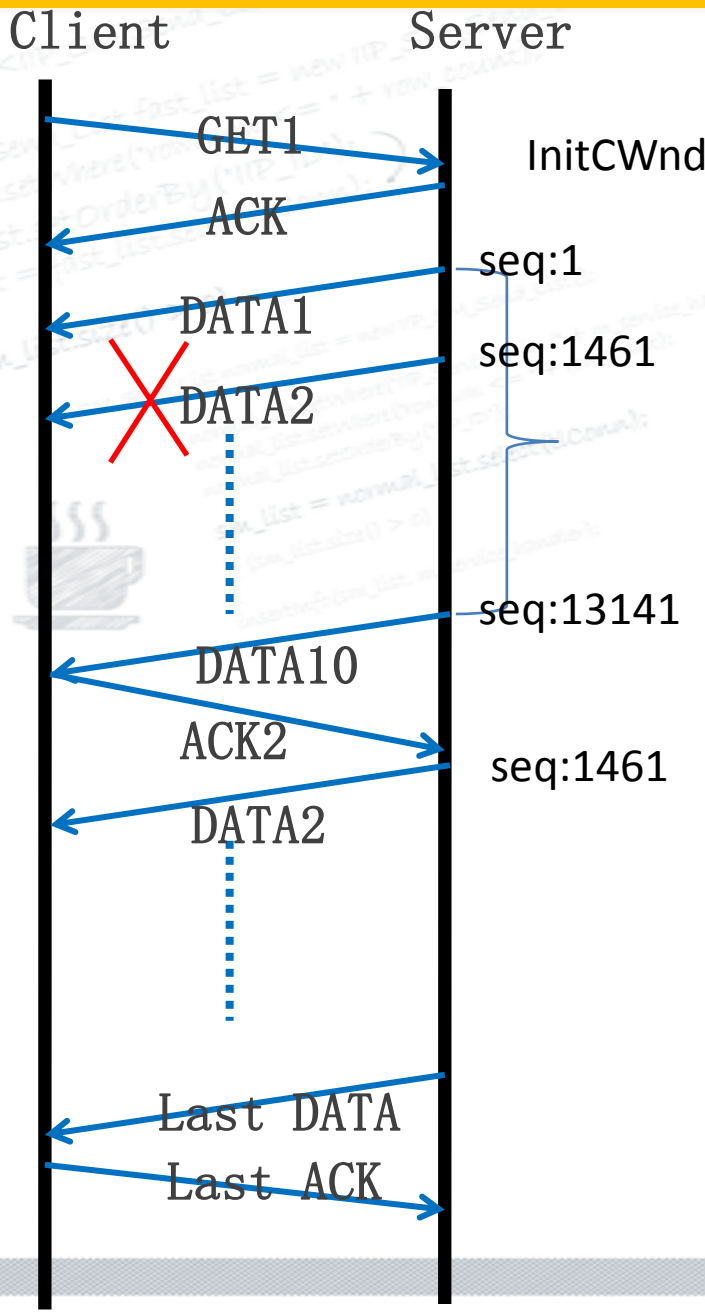
TCPFlow设计考虑 – 服务质量参数计算考量-RT计算，GET结束



Q1 RT两种计算方式哪种更精确？
方法1还是方法2

Q2 如何判断GET请求结束？
FIN？

TCPFlow设计考虑 – 服务质量参数计算考量 – GSO、TSO影响



1乱序丢包的判断？
2丢包率统计？
3 对象大小的统计？

开启GSO，TSO时，
抓包看到的是一个包

new_seq <= seq:1?

TCPFlow设计考虑 – 性能优化

- 优化
  - VIP查找： hash -> bitmap
  - state查找：桶个数， hash函数选择
  - 性能热点改进： sprintf，inet_ntoa等
  - 减少系统调用次数： 日志缓存
  - 减少内存使用： state结构体简化

TCPFlow未来展望

- 高性能抓包
- 支持更多业务类型
- 即将开源

结束语

- 团队成员新浪微博：
- @吴佳明_普空， @淘南坡， @RyanY_泠茗， @勿虚子，@Cz昂，@淘德泰

Q&A