网络基础知识快速入门实战解析

2017年11月15日，周三晚8点30分。在传统IT做过基于网络IP层和链路层的协议优化，目前在互联网公司做基于应用层和传输层的网络优化的宋璐带来了主题为《如何快速入门网络基础知识（TCP/IP 和 HTTP）》的交流。以下是主持人hrshy整理的问题精华，记录了作者和读者间问答的精彩片段。

内容提要：

- http和https有什么区别？
- http如何传输视频文件或者视频流？
- 理解tcp ip对编程，对日常工作的作用体现在什么地方？
- 怎么防止web网络抓包，使用https是不就可以防抓？
- 集群对单个服务器的心跳机制，常用的协议是什么？使用本专题介绍的协议是否可行？如果可以其优势体现在哪里，不行的话缺点是什么？
- QUIC主要应用场景有哪些？相比于这些场景以前的解决方案有何优势？
- 请问学习http相关知识的学习，对于其他语言比如java等从业者是否有必要？会有哪些影响？
- tcp、ip、http和socket之间的联系是什么？
- 全栈https的部署有什么难点吗？为什么建设银行仅有登录做了https 而不是全栈https？
- tcp和udp的区别有哪些？
- 请问TCP/IP对于测试人员需要掌握哪些知识点？
- DNS反劫持都有哪些措施？
- 可靠ＵＤＰ的重送机制如何设计在网路壅塞的时候效果会比较好？
- 网路状况的好坏是如何判定的？
- 请问chromium优化是如何进行网络优化的？
- 麻烦讲一下数字签名、数字证书、SSL可以吗？
- 很多人都觉得TCP/IP艰深难懂，请问你是怎么样学习TCP/IP的？
- https对服务器资源消耗怎么样？

问：**http**和**https**有什么区别？

答：可以简单认为https = http + tls。https是一种安全的加密传输协议，而http则是不加密的协议，因此https更加安全

在我们平时访问中，也可以看到浏览器的左上角的提示。

目前大部分网站都已经全量HTTPS了。

问：**http**如何传输视频文件或者视频流？

答： 对于流媒体传输，如果用http的话，一般是用hls协议，也就是HTTP Live Streaming协议。HLS协议是一个由苹果公司提出的基于HTTP的流媒体网络传输协议，它的原理是讲整个流分成一个个小的模块进行传输和下载的。

问：理解**TCP/IP**对编程，对日常工作的作用体现在什么地方？

答： 对于解决网络相关问题非常有用，一些比较常见的内核参数优化，比如我们要优化服务器的time-wait。

`tcp_tw_reuse = 1` 表示开启重用。允许将time-wait sockets重新用于新的TCP连接，默认为0，表示关闭；

`tcp_tw_recycle = 1` 表示开启TCP连接中time-wait sockets的快速回收，默认为0，表示关闭。

这样的例子还有很多，一般服务端的内核参数，如果没有一定的TCP/IP基础的话，很难准确地去理解，更不用说去优化了，并且我们也注意到，在很多公司的研发岗位招聘要求中，通常都要求应聘者具备TCP/IP方面的知识，甚至是精通。

问：怎么防止**Web**网络抓包，使用**https**是不就可以防抓？

答： HTTPS可以有效防止网络恶意抓包，但是由于成本原因，或者说HTTPS请求失败后，我们不可能完全全量的使用HTTPS，这个时候我们就需要在客户端对报文进行加密，然后在LB层或服务端进行解密。

问：集群对单个服务器的心跳机制，常用的协议是什么？使用本专题介绍的协议是否可行？如果可以其优势体现在哪里，不行的话缺点是什么？

答： 其实TCP已经有心跳机制，在一个集群下的不同服务之间也是通过这个心跳机制来实现通信，保证服务的可用性。之前在华为工作时，路由器集群之间的心跳报文是华为自己定制的一套体系，为了保证路由表和MAC表的正常转发，多台路由器之间存在心跳机制，这个机制是建立在IP层面上的，但是不具有通用性。

问：**QUIC**主要应用场景有哪些？相比于这些场景以前的解决方案有何优势？

答： 目前谷歌已经开始广泛上quic，前不久腾讯云也宣布拥抱变化，支持quic。Google在YouTube上部署后有数据显示，视频缓冲卡顿少了30%，用户体验得到了非常大的提升，其它一些已经部署了QUIC协议的网站，传输速度也有一定的提升。

QUIC相比传统的HTTP传输，有以下的优势：

1. 建立连接快，除了首次建立连接需要1-RTT之外，其余的连接建立均是0—RTT，别小看一个RTT（往返时延），如果在弱网环境下，也要几百毫秒，这已经是一个非常大的优化了。

2. 采用UDP传输，UDP速度比TCP要快，同时QUIC还提供像TCP一样的稳定传输。

3. 采用流传输，当用户的五元组(目的ip、源ip、目的端口、源端口和上层协议)任一一个发生变化时，不需要再重新建立连接，QUCI可以直接进行切换，节省了不少时间。

问：请问学习**http**相关知识的学习，对于其他语言比如**java**等从业者是否有必要？会有哪些影响？

答： 首先语言和网络知识的学习并不冲突，拿武功来说，语言相当于是一门招式，而网络知识则是内功基础，试想一下，如果你作为一个后端java开始，遇上网络问题却一窍不通，那处理问题起来会多么僵硬。

问：**tcp**、**ip**、**http**和**socket**之间的联系是什么？

答： ip是网络层协议，上层的传输层是tcp，在上层的应用层协议是http，这是最为常见的一种报文形式了，在我们的文章中，也提到了，你去抓一个http报文，其层次就是这样的。

但实际上http协议并没有规定其传输层协议一定是tcp，你可以用udp封装进行传

输，理论上行得通，但为什么不这么做？这是因为udp不提供可靠的传输服务，而tcp提供。而socket可以看做是源ip、目的ip、源端口、目的端口以及所在协议的五元组，平时我们也有关于socket的编程，这个时候实际上就是你提供这个五元组内的信息，再调用系统提供的socket接口，便可以完成服务端和客户端的传输。

问：全栈**https**的部署有什么难点吗？为什么建设银行仅有登录做了**https**而不是全栈**https**？

答： 全栈https最大的问题就在于成本，因为https协议需要服务器或者云端的支持，并且加解密也存在资源上的大量开销。我们文章中，某些网站之所以没有全量https，是因为有些业务不是那么看重安全性或者说这部分业务及时被截获也影响不大，一般很多网站或app的log都还是采用http传输。

问：**tcp**和**udp**的区别有哪些？

答： 简单点讲，udp协议更为轻量，不提供可靠的传输服务。而tcp则提供可靠的，基于连接的字节流服务，因此它的报文头也更长，一般为20字节，而udp只有8字节。再延伸下，quic协议采用udp协议，也是看中udp更为轻量和便捷，当然quic在应用层做了传输保证，因此其才具备又快又准的特性。

问：请问 **TCP/IP** 对于测试人员需要掌握哪些知识点？

答： 测试人员的话：

1. 抓包是必须的，各种抓包软件必须使用的十分熟练；
2. 会看报文，抓包完以后，要会分析报文，至少要熟悉http报文头的一些关键字段都是什么意思；
3. 能够熟练掌握tcp状态迁移图，也就是文章中提到的，对于连接的建立和断开过程要很熟练，必要时还要上开发机看这些连接的状态。

问：**DNS**反劫持都有哪些措施呢？

答： 一般来说DNS反劫持有几种有效的做法：

1. 域名对应多个IP地址，这样即使一个IP被劫持，也还有一个备用的IP地址，

不至于导致服务完全瘫痪。

2. 利用HTTPDNS替代系统自带的DNS解析，HTTPDNS是近几年比较常用的DNS反劫持手段，顾名思义，它是用HTTP协议来进行DNS解析的，这样做的好处有：HTTP协议传输层采用的是TCP协议，而系统自带的DNS解析走的则是UDP协议，TCP比UDP靠谱很多，另外HTTP协议可以做到更为精准的调度和即时性。之前提到DNS解析中有一个重要参数TTL，也就是说如果你想修改DNS解析结果，通过传统改IP的方式，解析结果必须要在TTL后才能生效，而HTTPDNS则灵活很多。

目前提供HTTPDNS服务的厂商中，最大的当属阿里云和DNSPod了，有这方面需求的同学可以关注一下。

问：可靠ＵＤＰ的重送机制如何设计在网路壅塞的时候效果会比较好？

答： 重传机制的话，有以下几个思路：

1. 利用tcp的指数退避，在需要重传时，根据重传次数，通过1s，2s，4s这样的方式依次类推，来进行重传；
2. 上述方式存在一定的弊端，那就是没有考虑网络的情况，特别是在弱网或者是网络情况较差的情况下，反复重试，实际上只会更加加剧报文的拥塞，因此在设计重传时，一定要考虑网络情况，简单点说就是动态监测当前带宽占用率，根据监测数据，来选取重传的时间间隔和时机。

问：网路状况的好坏是如何判定的？

答： 网络情况的监测实际上还是比较困难的，利用ping的方式，也只能是看到对于某个ip的访问延迟而已。如果是服务器端的网络情况，建议可以利用一些监控平台，来实时进行监控。而对于客户端的网络情况，目前我们是通过利用chromium内核中的net库中的日志记录模块，定点采样用户的请求过程，记录每一步的网络情况和用时，然后再进行分析和处理。

问：请问**chromium**优化是如何进行网络优化的？

答： chromium是chrome浏览器背后的内核引擎，其中有专门的net库来进行网络处理和优化，就拿DNS来说，chromium并不是直接采用底层的dns解析函数，而是自己通过进程调度来实现这一过程，还有就是chromium支持多种协议，之前提到

的quic也是chrome发明，自然在chromium中也有体现，同时还有上一题中提到的连接日志，可以详细记录一个完整请求下的各种过程的耗时，比如说DNS解析，tcp连接以及tls握手等等。

问：麻烦你讲一下数字签名、数字证书、**SSL**可以吗？

答： 好的，数字签名简单点说就是传输过程中，发送者提供的一种不可伪造的证明。而数字证书，我们在访问一些网站时，会要求我们下载证书才能访问，相当于就是一种身份的象征。SSL协议作为一种加密协议，包括握手，还有数据传输。

在HTTPS协议中的握手协议指的就是SSL握手，这里的过程大致是：

1. client_hello：客户端发起请求；
2. server_hello：服务端返回协商的信息结果；
3. 证书校验；
4. 合法性验证通过之后，客户端计算产生随机数字Pre-master，并用证书公钥加密，发送给服务器；
5. 服务器用私钥解密加密的Pre-master数据；
6. 握手结束。

问：很多人都觉得**TCP/IP**艰深难懂，请问你是怎么样学习**TCP/IP**的？

答： 我觉得是一个循序渐进的过程，主要是先从最基础的TCP/IP的书籍，然后就是尽量能够在实践中去利用。可能和我的工作经历有关，第一份工作就是在华为，接触的就是底层的二层转发和三层路由，所以必须去学习这些网络知识，目前这份工作也是做网络调度和客户端网络优化的。另外网络问题的处理，比如说一些线上问题，可以非常快速地帮你提升网络知识素养。

问：**https**对服务器资源消耗怎么样？

答： 我们目前https的解析是放在腾讯云和阿里云上面的，量大概是95%左右，主要是https增加了握手次数，并且握手失败以后，还需要fallback到http，其实更多的是这部分的处理。这里的话，本身就多了三次握手，2个RTT，而且移动端网络变化复杂，即便是长连接，也存在连接的反复建立，因此损耗还是不少的。

GitChat