

HTTP知识

一、HTTP的发展

1. 概念

- 万维网：World Wide Web (www)
- HTTP：超文本传输协议(HyperText Transfer Protocol)

1.3 基本概念及技术

- URI：即统一资源标识符，作为互联网上资源的唯一身
- HTML：即超文本标记语言，描述超文本文档；
- HTTP：即超文本传输协议，用来传输超文本。

2. HTTP/1.1 的特点

- 增加了 PUT、DELETE 等新的方法；
- 增加了缓存管理和控制；
- 明确了连接管理，允许持久连接；
- 允许响应数据分块（chunked），利于传输大文件
- 强制要求 Host 头，让互联网主机托管成为可能。

3. HTTP/2 的特点 - gRpc 【Google】

- 二进制协议，不再是纯文本；
- 可发起多个请求，废弃了 1.1 里的管道；
- 使用专用算法压缩头部，减少数据传输量；
- 允许服务器主动向客户端推送数据；
- 增强了安全性，“事实上”要求加密通信。

4. HTTP/3 - HTTP over QUIC 【Google】

二、深入理解HTTP

1. 什么是HTTP

HTTP 就是超文本传输协议，也就是 HyperText Transfer Protocol。

2. 超文本、传输、协议各自代表的含义

2.1 Http-协议

2.2 协议的特点

- 协议必须要有两个或多个参与者
- 协议是对参与者的一种行为约定和规范

HTTP 是一个用在计算机世界里的协议。它使用计算机能够理解的语言确立了一种计算机之间交流通信的规范，以及相关的各种控制和错误处理方式。

2.3 Http-传输

2.4 传输的特点

- HTTP 协议是一个“双向协议”
- 数据虽然是在 A 和 B 之间传输，但并没有限制只有 A 和 B 这两个角色，允许中间有“中转”或者“接力”。

HTTP 是一个在计算机世界里专门用来在两点之间传输数据的约定和规范。

2.5 Http-超文本

2.6 超文本的特点

它是文字、图片、音频和视频等的混合体，最关键的是含有“超链接”，能够从一个“超文本”跳跃到另一个“超文本”，形成复杂的非线性、网状的结构关系。

HTTP 是一个在计算机世界里专门在两点之间传输文字、图片、音频、视频等超文本数据的约定和规范

HTTP 通常跑在 TCP/IP 协议栈之上，依靠 IP 协议实现寻址和路由、TCP 协议实现可靠数据传输、DNS 协议实现域名查找、SSL/TLS 协议实现安全通信。此外，还有一些协议依赖于 HTTP，例如 WebSocket、HTTPDNS 等。这些协议相互交织，构成了一个协议网，而 HTTP 则处于中心地位。

HTTP 是一个用在计算机世界里的协议，它确立了一种计算机之间交流通信的规范，以及相关的各种控制和错误处理方式。

HTTP 专门用来在两点之间传输数据，不能用于广播、寻址或路由。

HTTP 传输的是文字、图片、音频、视频等超文本数据。

HTTP 是构建互联网的重要基础技术，它没有实体，依赖许多其他的技术来实现，但同时许多技术也都依赖于它。

3. TCP/IP

3.1 IP 协议的概念

IP协议是“Internet Protocol”的缩写，主要目的是解决寻址和路由问题，以及如何在两点间传送数据包。IP 协议使用“IP 地址”的概念来定位互联网上的每一台计算机。可以对比一下现实中的电话系统，你拿着的手机相当于互联网上的计算机，而要打电话就必须接入电话网，由通信公司给你分配一个号码，这个号码就相当于 IP 地址。

3.2 TCP 协议的概念

TCP 协议是“Transmission Control Protocol”的缩写，意思是“传输控制协议”，它位于 IP 协议之上，基于 IP 协议提供可靠的、字节流形式的通信，是 HTTP 协议得以实现的基础。

TCP协议的特点：

- 可靠：保证数据不丢失
- 字节流：保证数据的完整性

3.3 URI、URL

- 3.3.1 什么是URI?

URI (Uniform Resource Identifier) ， 统一资源标识符，使用它能够唯一地标记互联网上资源。(以：<http://nginx.org/en/download.html> 为例)

- 3.3.2 URI 基本的部分构成
 - - 协议名：即访问该资源应当使用的协议，如http
 - - 主机名：即互联网上主机的标记，可以是域名或 IP 地址，如 baidu.com
 - - 路径：即资源在主机上的位置，使用“/”分隔多级目录，在这里是“/en/download.html”
- 3.4 什么是URL?

URL (Uniform Resource Locator) ， 统一资源定位符，也就是我们俗称的“网址”，它实际上是 URI 的一个子集，不过因为这两者几乎是相同的，差异不大，所以通常不会做严格的区分。

4. HTTPS

4.1 什么是HTTPS?

运行在 SSL/TLS 协议上的 HTTP

4.2 什么是SSL/TLS?

它是一个负责加密通信的安全协议，建立在 TCP/IP 之上，所以也是个可靠的传输协议，可以被用作 HTTP 的下层。

HTTPS <===> HTTP+SSL/TLS+TCP/IP

SSL: Secure Socket Layer

TLS: Transport Layer Security

TLS是SSL的3.0标准化版本

5. 代理

5.2 代理的概念

代理 (Proxy) 是 HTTP 协议中请求方和应答方中间的一个环节, 作为“中转站”, 既可以转发客户端的请求, 也可以转发服务器的应答。

5.2 代理的种类

- 匿名代理: 完全“隐匿”了被代理的机器, 外界看到的只是代理服务器;
- 透明代理: 顾名思义, 它在传输过程中是“透明开放”的, 外界既知道代理, 也知道客户端;
- 正向代理: 靠近客户端, 代表客户端向服务器发送请求;
- 反向代理: 靠近服务器端, 代表服务器响应客户端的请求;

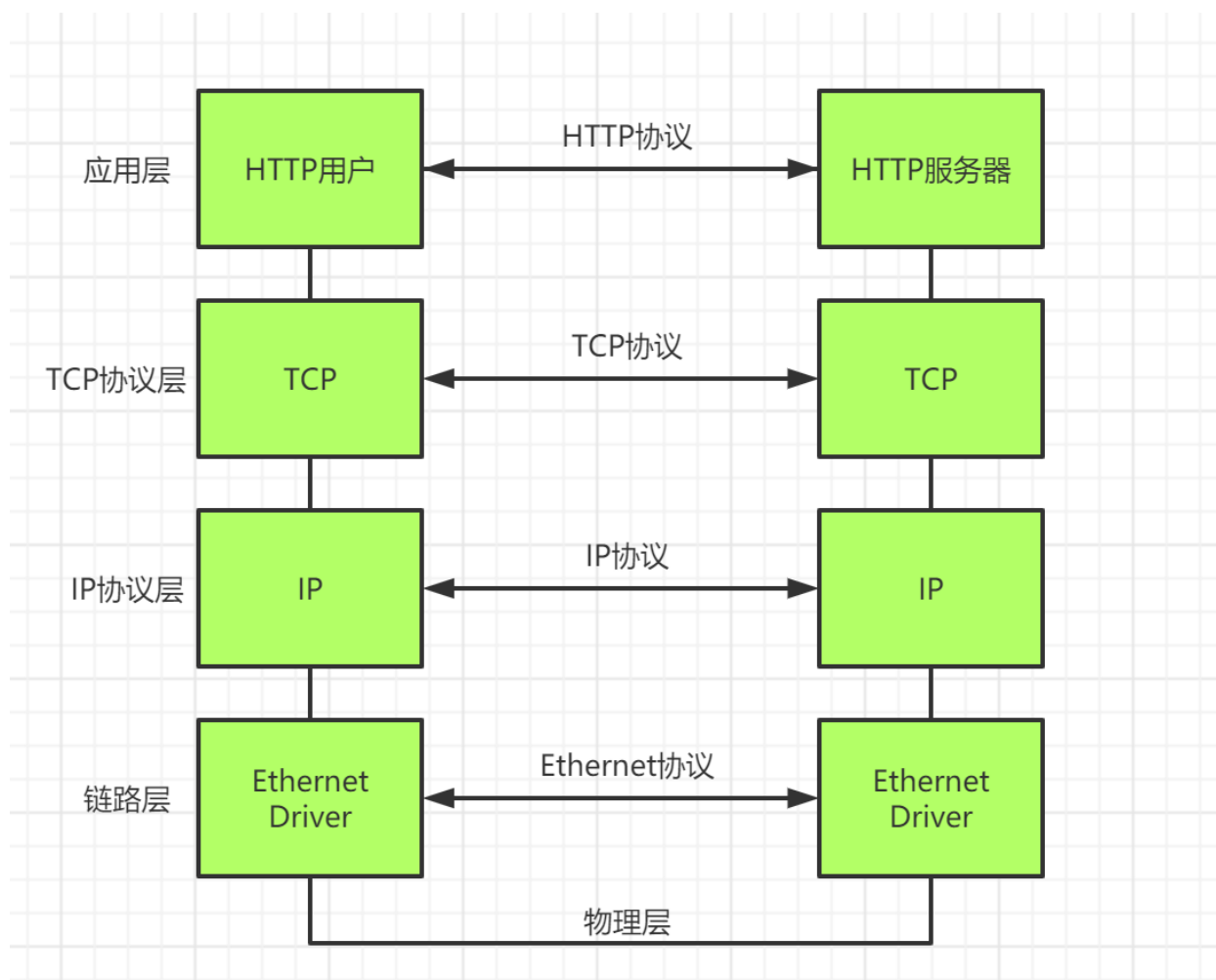
CDN也是一种代理, 其扮演的角色: 透明代理和反向代理的角色

5.3 代理的作用

- 负载均衡: 把访问请求均匀分散到多台机器, 实现访问集群化;
- 内容缓存: 暂存上下行的数据, 减轻后端的压力; 安全防护:
- 隐匿 IP, 使用 WAF 等工具抵御网络攻击, 保护被代理的机器;
- 数据处理: 提供压缩、加密等额外的功能。

-
- TCP/IP 是网络世界最常用的协议, HTTP 通常运行在 TCP/IP 提供的可靠传输基础上
 - DNS 域名是 IP 地址的等价替代, 需要用域名解析实现到 IP 地址的映射;
 - URI 是用来标记互联网上资源的一个名字, 由“协议名 + 主机名 + 路径”构成, 俗称 URL;
 - HTTPS 相当于“HTTP+SSL/TLS+TCP/IP”, 为 HTTP 套了一个安全的外壳;
 - 代理是 HTTP 传输过程中的“中转站”, 可以实现缓存加速、负载均衡等功能。
-

5. TCP\IP协议层



5.1 名词解释 (从小往上)

- 第一层叫“链路层” (link layer)，负责在以太网、WiFi 这样的底层网络上发送原始数据包，工作在网卡这个层次，使用 MAC 地址来标记网络上的设备，所以有时候也叫 MAC 层。
- 第二层叫“网际层”或者“网络互连层” (internet layer)，IP 协议就处在这一层。因为 IP 协议定义了“IP 地址”的概念，所以就可以在“链路层”的基础上，用 IP 地址取代 MAC 地址，把许许多多的局域网、广域网连接成一个虚拟的巨大网络，在这个网络里找设备时只要把 IP 地址再“翻译”成 MAC 地址就可以了。
- 第三层叫“传输层” (transport layer)，这个层次协议的职责是保证数据在 IP 地址标记的两点之间“可靠”地传输，是 TCP 协议工作的层次。TCP 是一个有状态的协议，需要先与对方建立连接然后才能发送数据，而且保证数据不丢失不重复。
- UDP 则比较简单，它无状态，不用事先建立连接就可以任意发送数据，但不保证数据一定会发到对方。两个协议的另一个重要区别在于数据的形式。
- TCP 的数据是连续的“字节流”，有先后顺序，而 UDP 则是分散的小数据包，是顺序发，乱序收。
- 第四层叫“应用层” (application layer)，有各种面向具体应用的协议。例如 Telnet、SSH、FTP、SMTP、HTTP。

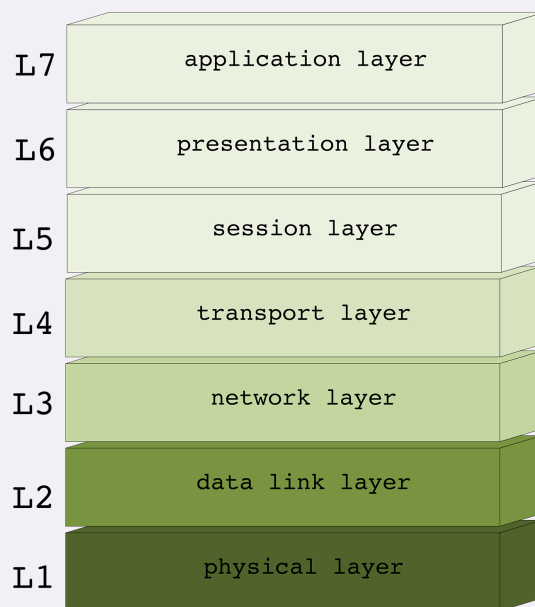
5.2 各层传输单位

- MAC 层的传输单位是帧 (frame)
- IP 层的传输单位是包 (packet)
- TCP 层的传输单位是段 (segment)
- HTTP 的传输单位则是消息或报文 (message)

6. OSI 网络分层模型

6.1 OSI概念：

OSI，全称是“开放式系统互联通信参考模型” (Open System Interconnection Reference Model)



6.2 各层说明

- 第一层：物理层，网络的物理形式，例如电缆、光纤、网卡、集线器等等；
- 第二层：数据链路层，它基本相当于 TCP/IP 的链接层；
- 第三层：网络层，相当于 TCP/IP 里的网际层；
- 第四层：传输层，相当于 TCP/IP 里的传输层；
- 第五层：会话层，维护网络中的连接状态，即保持会话和同步；
- 第六层：表示层，把数据转换为合适、可理解的语法和语义；
- 第七层：应用层，面向具体的应用传输数据。

凡是由操作系统负责处理的就是四层或四层以下，否则，凡是需要由应用程序（也就是你自己写代码）负责处理的就是七层。

二层转发：设备工作在链路层，帧在经过交换机设备时，检查帧的头部信息，拿到目标mac地址，进行本地转发和广播。

三层路由：设备工作在ip层，报文经过有路由功能的设备时，设备分析报文中的头部信息，拿到ip地址，根据网段范围，进行本地转发或选择下一个网关

dns：网络请求的第一步是域名解析，所以工作在应用层

cdn：应用层

7. DNS详解

7.1 DNS核心系统的结构：

是一个三层的树状、分布式服务，基本对应域名的结构

- 根域名服务器（Root DNS Server）：管理顶级域名服务器，返回“com” “net” “cn”等顶级域名服务器的IP地址；
- 顶级域名服务器（Top-level DNS Server）：管理各自域名下的权威域名服务器，比如com顶级域名服务器可以返回apple.com域名服务器的IP地址；
- 权威域名服务器（Authoritative DNS Server）：管理自己域名下主机的IP地址，比如apple.com权威域名服务器可以返回www.apple.com的IP地址。

7.2 域名解析

你要访问“www.apple.com”，就要进行下面的三次查询：

访问根域名服务器，它会告诉你“com”顶级域名服务器的地址；

访问“com”顶级域名服务器，它再告诉你“apple.com”域名服务器的地址；

最后访问“apple.com”域名服务器，就得到了“www.apple.com”的地址。

7.3 DNS请求方式

- 递归查询
- 迭代查询

7.4 DNS解析的过程

浏览器DNS缓存->操作系统缓存->Hosts文件->非权威域名服务器->根域名服务器->顶级域名服务器->权威域名服务器。一直对浏览器/本机如何知道根服务器地址存在疑惑。

三、HTTP协议进阶

1. HTTP 协议请求全过程

1.1 浏览器 HTTP 请求过程：

- 浏览器从地址栏的输入中获得服务器的 IP 地址和端口号；
- 浏览器用 TCP 的三次握手与服务器建立连接；
- 浏览器向服务器发送拼好的报文；
- 服务器收到报文后处理请求，同样拼好报文再发给浏览器；
- 浏览器解析报文，渲染输出页面。

1.2 HTTP 协议请求 - 应答的全过程

- HTTP 协议基于底层的 TCP/IP 协议，所以必须要用 IP 地址建立连接；
- 如果不知道 IP 地址，就要用 DNS 协议去解析得到 IP 地址，否则就会连接失败；
- 建立 TCP 连接后会顺序收发数据，请求方和应答方都必须依据 HTTP 规范构建和解析报文；
- 为了减少响应时间，整个过程中的每一个环节都会有缓存，能够实现“短路”操作；
- 虽然现实中的 HTTP 传输过程非常复杂，但理论上仍然可以简化成实验里的“两点”模型。

