

## 图解 HTTP — (了解 web 及网络基础)

享受着爱和荣誉的人，才会感到生活的乐趣。—— 夏吉尔硕

### 起步：

关于 HTTP 的背景及从 1.0 开始的演变历史就不过多阐述，有兴趣的可以自行了解，本章内容及进度根据《图解 HTTP》图灵系列丛书，加上自己搜索的互联网资料补充总结，需要这本书的 PDF 版，可留邮箱。

### TCP/IP 的分层

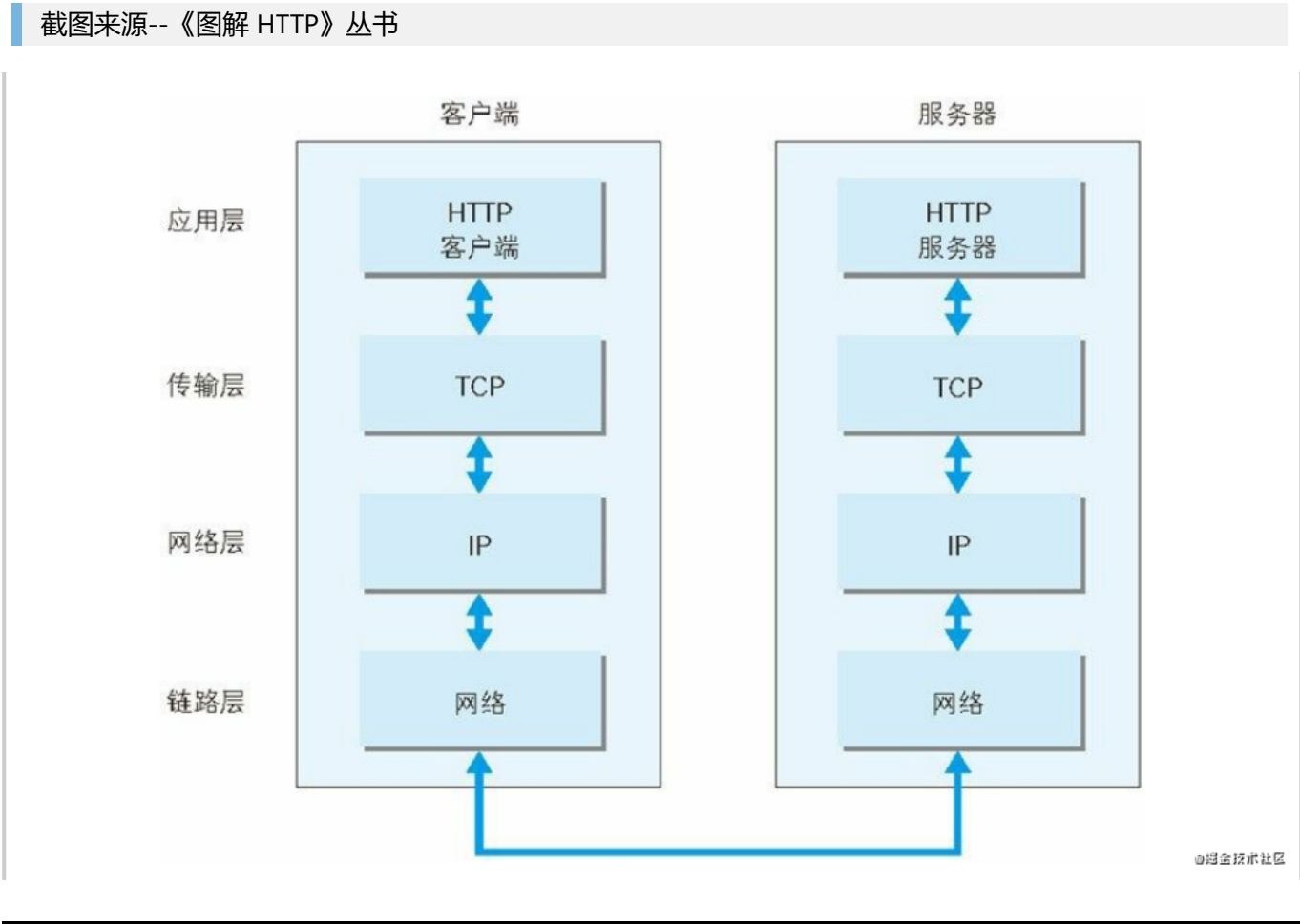
TCP/IP 协议族里重要的一点就是分层。TCP/IP 协议族按层次分别分为 4 层：

- 应用层
  - 应用层决定了向用户提供应用服务时通信的活动。
  - 比如，**FTP**（文件传输协议）和 **DNS**（域名系统）服务及 **HTTP** 协议
- 传输层
  - 传输层对上层应用层，提供处于网络连接中的两台计算机之间的数据传输。
  - 传输层有两个性质不同的协议：**TCP**（传输控制协议）和 **UDP**（用户数据报协议）
    - 二者具有很大的区别，需重点掌握
      - 连接方面：
        - TCP 面向连接，即必须先建立安全连接（三次握手）才能发送数据
        - UDP 无连接，发送数据前不需要建立连接
      - 安全方面：
        - TCP 安全可靠，无差错，不丢失，不重复且按序。
        - UDP 无连接不可靠，尽力进行数据发送，但丢包风险高，不保证数据顺序
      - 连接对象数量方面
        - TCP 仅支持一对一通信
        - UDP 支持一对一，一对多，多对多通信对象
      - 性能方面：
        - TCP 信息包较长，有 20-60 个字节，开销较大
        - UDP 信息包标题短，只有 8 个字节
      - 有序性
        - TCP 利用 seq 序列号对包进行排序，UDP 不排序。
      - 使用场景：
        - TCP 适用于效率要求相对较低，准确性可靠传高的传输，例如文件传输
        - UDP 适用于效率要求高，准确性相对低的场景，比如（QQ 聊天，IP 电话、视频会议、直播啥的）
  - 网络层
    - 网络层用来处理在网络上流动的数据包。
      - 数据包是网络传输的最小数据单位
      - 网络层为数据包选择最佳路径，使得数据包达到目的地
      - 补充：会出现丢包的情况，但是丢包由传输层管理，网络层不会去管理。
  - 链路层
    - 链路层用来处理连接网络的硬件部分。

- 链路(物理链路): 从一个结点到相邻结点的一段物理线路，而中间没有任何其他的交换结点。
- 透明传输: 表示无论什么样的比特组合的数据，都能按照原样没有差错的通过这个数据链路层。
- 封装成帧: 帧是数据链路层的传送单位。帧长等于帧的数据部分加上帧的首尾部长度。首部和尾部（确定帧的界限）。
- 差错检验: 实现无比特差错。

## TCP/IP 通信传输流

通信传输流程图，如下所示：

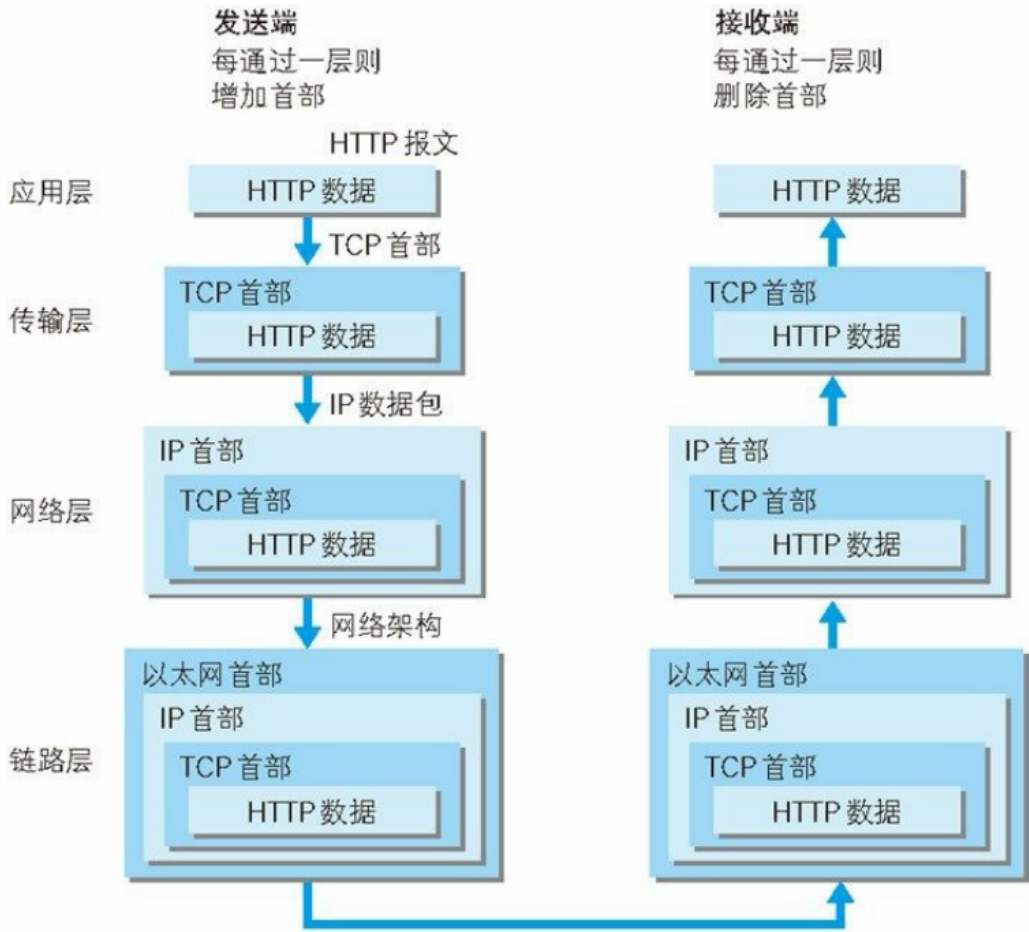


- 利用TCP/IP协议族网络通信时，通过分层顺序与对方通信
  - 发送方：从应用层往下进行
  - 接收方：从下往应用层进行
- 举例 HTTP,客户端作为发送端，请求某个 web 页面数据
  - 首先传输层（TCP协议）
    - 把从应用层接受的数据（HTTP 请求报文）分割
    - 在各个报文上打上标记序号（TCP 具有按序性）及端口号
    - 转发给网络层
  - 其次到网络层（IP协议）
    - 在传输层整理完的数据基础上，增加通信目的地的 MAC地址
    - 转发给链路层
  - 然后到接收端的链路层 - 接收数据，按顺序往上层发送 - 传输到服务器端的应用层时，服务器才接收到客户端发送来的 HTTP 请求

客户端发送的请求数据，在经历层层传输时，势必会被每层所属的首部信息包裹，这叫做**封装**。同理，从链路层往服务器端的应用层向上传输时，每层对应的首部信息也会层层剥去。

- 如下图所示：

截图来源--《图解 HTTP》丛书。



© 掘金技术社区

## 与 HTTP 密切相关的协议：IP、TCP 和 DNS

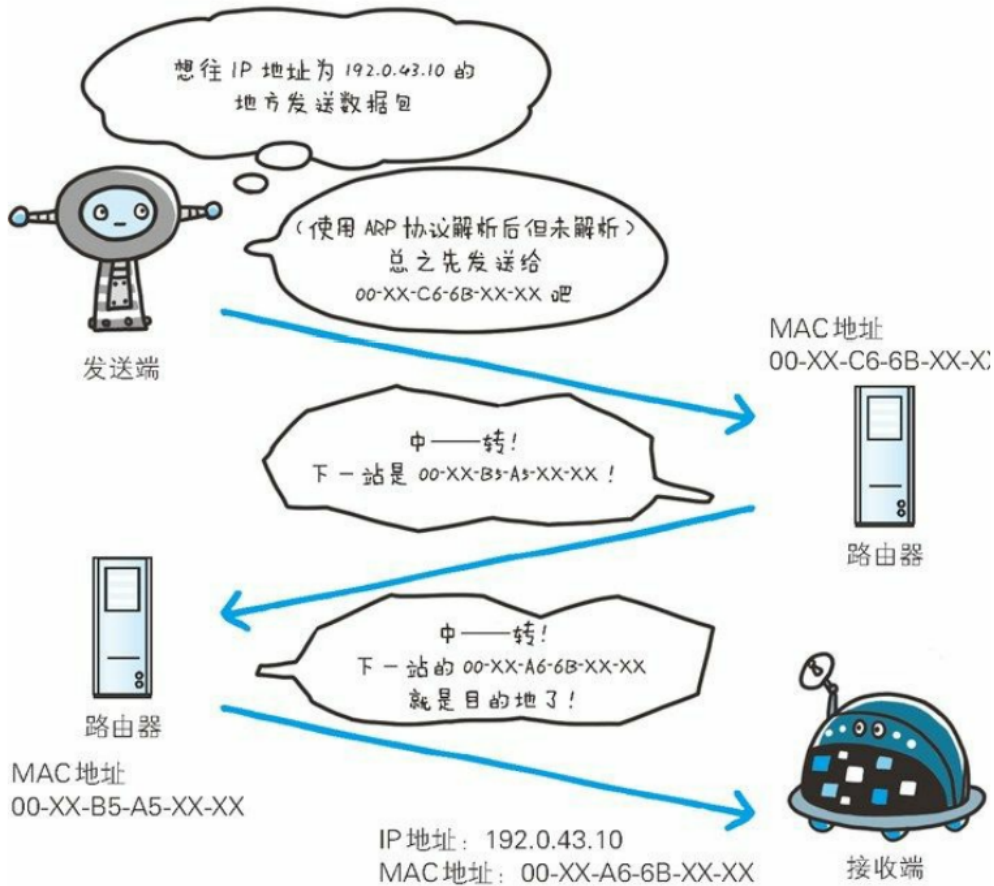
### IP 协议：

- 负责传输的 IP 协议 概念：
  - 几乎所有的使用网络的系统都会用到 IP 协议，重要性可见一斑
  - TCP/IP 协议族中的 IP 指的是 **网际协议**
  - **IP (Internet Protocol)** 网际协议位于 **网络层**
- IP 协议的作用：
  - 把各种数据包传送给对方
  - 保证必须传送到对方那里，需满足各类条件
    - 重要条件之一：**IP 地址**
    - 重要条件之二：MAC 地址 (Media Access Control Address) 。

IP 地址指明了节点被分配的地址。MAC 地址是指网卡所属的固定地址。IP 地址可以和 MAC 地址进行配对 **IP 地址可变**， **MAC地址基本上不会更改**。\*

- 使用 ARP 协议凭借 MAC 地址进行通信
  - IP 间的通信依赖 MAC 地址
  - 网络上，通信双方通常经过多台计算机和网络设备中转才连接到对方
  - 中转时，利用下一站中转设备的 MAC 地址搜索下一中转目标
  - 此时会使用 ARP 协议
    - ARP 是用以解析地址的协议，根据通信方的 IP 地址反查出对应的 MAC 地址。
- 如下图所示：

截图来源--《图解 HTTP》丛书



© 掘金技术社区

## TCP 协议：

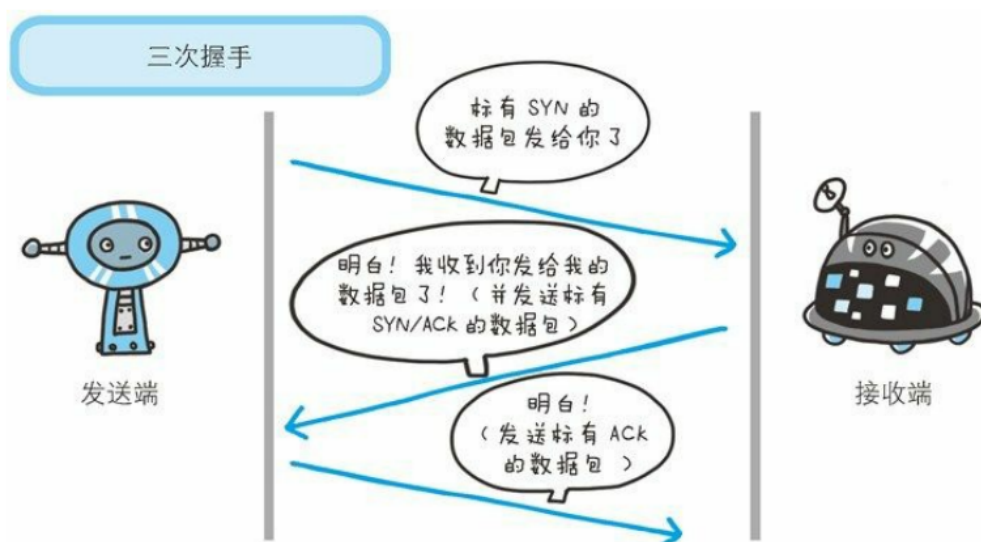
- 确保可靠的 TCP 协议 概念：
  - 字节流服务：便于传输，把数据分割成以报文段为单位的数据包
  - TCP 协议提供字节流服务，为了便于传送大体系数据
  - TCP 协议能确保数据到达目标
  - TCP 协议位于 **传输层**
- 如何确保数据到达目标？
  - TCP 协议使用了**三次握手 (three-way handshaking)** 策略
  - TCP 将数据包传送后，会向对方确认是否成功 (UDP 不会)
- 知晓一些标识符

- SYN: 同步标志
- ACK: 确认标志
- FIN: 结束标志

'握手' 中使用 TCP 的标志—— SYN (synchronize) 和 ACK (acknowledgement)

- 1. 发送端先发送一个带 SYN 标志的数据包给对方。
- 2. 接收端收到后, 回传一个带有 SYN/ACK 标志的数据包以示传达确认信息
- 3. 最后, 发送端再回传一个带 ACK 标志的数据包, 代表“握手”结束。
- 补充: 若上述任意过程中断, 则重新来过
- 如下图所示:

截图来源--《图解 HTTP》丛书



© 掘金技术社区

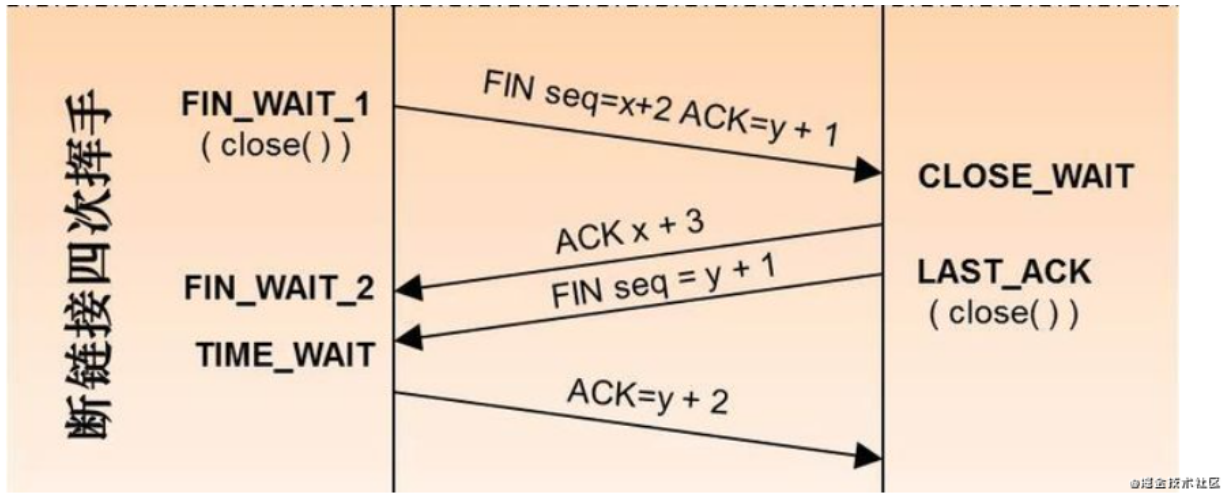
通俗来描述:

- A: 客户端先向服务端发送一个 SYN 包告诉它自己的初始序列号是 X
- B: 服务端收到 SYN 包后回复给客户端一个 ACK 确认包, 表明自己收到了, 并且也发送一个 SYN 包告诉客户端自己的初始序列号是 Y;
- C: 客户端收到这两包后, 会回复服务端一个 ACK 确认包表明收到。

'挥手': 指 TCP 断开连接, 停止数据的双向传输, 并且回收资源

- 如下图所示:

截图来源--《腾讯技术工程》公众号



通俗来描述：

- A：客户端发送一个FIN包，告知服务端，没有数据传输了
- B：服务端收到后回ACK确认包，表明知道了；
- C：然后服务端发送FIN 包告诉客户端，也没有数据传输了
- D：客户端回一个ACK确认包，表明知道了
- 至此四次挥手完成，TCP 断开连接。

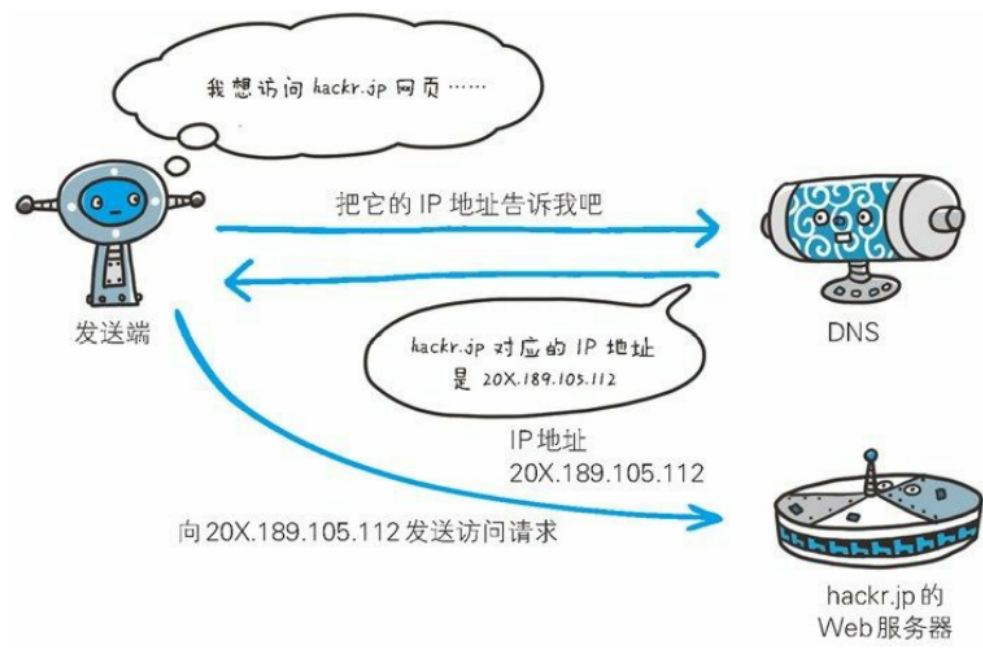
补充：四次挥手可否变成三次呢？若服务端收到 客户端 的 FIN 结束标志包告知没有数据传输时，自己也没数据需要发送给 客户端，则对 客户端 的 ACK 确认包和 服务端 自己的 FIN 结束标志包就可以合并成为一个包发送过去，这样四次挥手就可以变成三次了

DNS 服务：

- 负责域名解析的 DNS 服务 概念：
  - DNS (Domain Name System) 服务：提供域名到 IP 地址之间的解析服务。
  - IP 地址访问不适用于人类记忆，从而通常使用域名访问
  - DNS 将域名和 IP 地址相互映射为一个分布式数据库，便于访问互联网
  - DNS 协议和 HTTP 协议一样位于 应用层
- 如下图所示 DNS协议提供通过域名查IP，或反向从IP反查域名服务：

截图来源--《图解 HTTP》丛书)

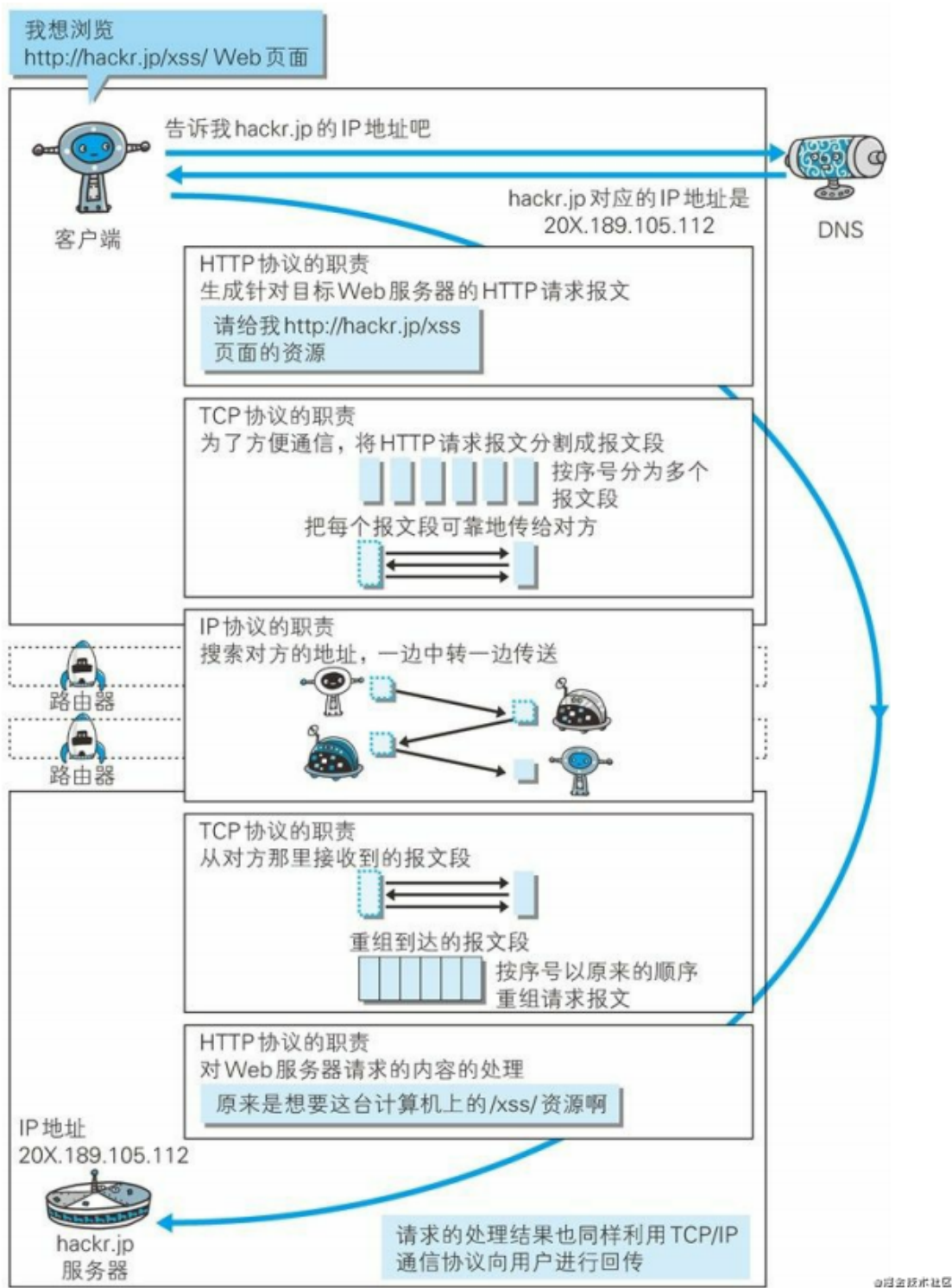




各协议与 HTTP 协议的关系及发挥的作用

- 如下图所示各协议之间协同合作，完成通信及数据交换：

截图来源--《图解 HTTP》丛书)



## URL 和 URI

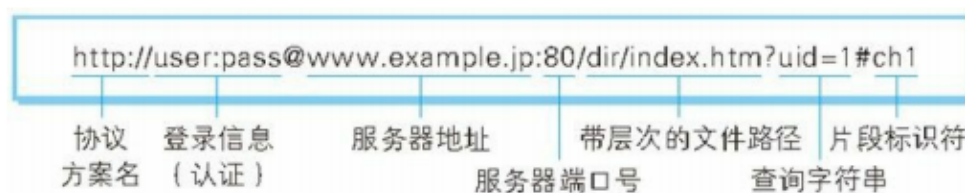
- URL 和 URI 概念:

- URL 为统一资源 **定位符** ---URI 为统一资源 **标识符**
- URI 分为三种, URL or URN or (URL and URI) 目前所有的 URI 几乎都是 URL
- URL 可以看作是 URI 的子集。
- 浏览器访问 web 页面时, 需要输入的网页地址认为是 URL (不是废话)

- 统一资源定位符 URL



- URL 是 **Uniform Resource Locator** 的缩写
  - URL 由三部分组成：资源类型、存放资源的主机域名、资源文件名
  - (也可认为由 4 部分组成：协议、主机、端口、路径)
- 统一资源标识符号 **URI**
  - URI 是 **Uniform Resource Identifier** 的缩写
    - **Uniform** 指统一的格式，便于处理不同类型的资源
    - **Resource** 指可标识的任何东西，大多数都可以作为资源
    - **Identifier** 指可标识的对象，也称为标识符
  - 综述：**URI 是某个协议方案标识的资源的定位标识符**
    - 协议方案：访问资源所使用的协议类型名称
    - 比如：使用 HTTP (HTTPS) 时，协议方案就是 http(https)
  - 绝对的 URI 格式：



© 掘金技术社区

- 知乎示例
  - URI：我的身份证号
  - URL：人类协议://中国/河南省/郑州市/光明路/3 栋 4 号/小土豆人

## 总结：

URI 为标识符，URL 为定位符，两者都可以确定唯一资源，其中 **通过地址规则实现的 URI 可以被称作 URL**，**URL 是 URI 的一种表现，所以 URI 作为更宽泛的定义包含了 URL**