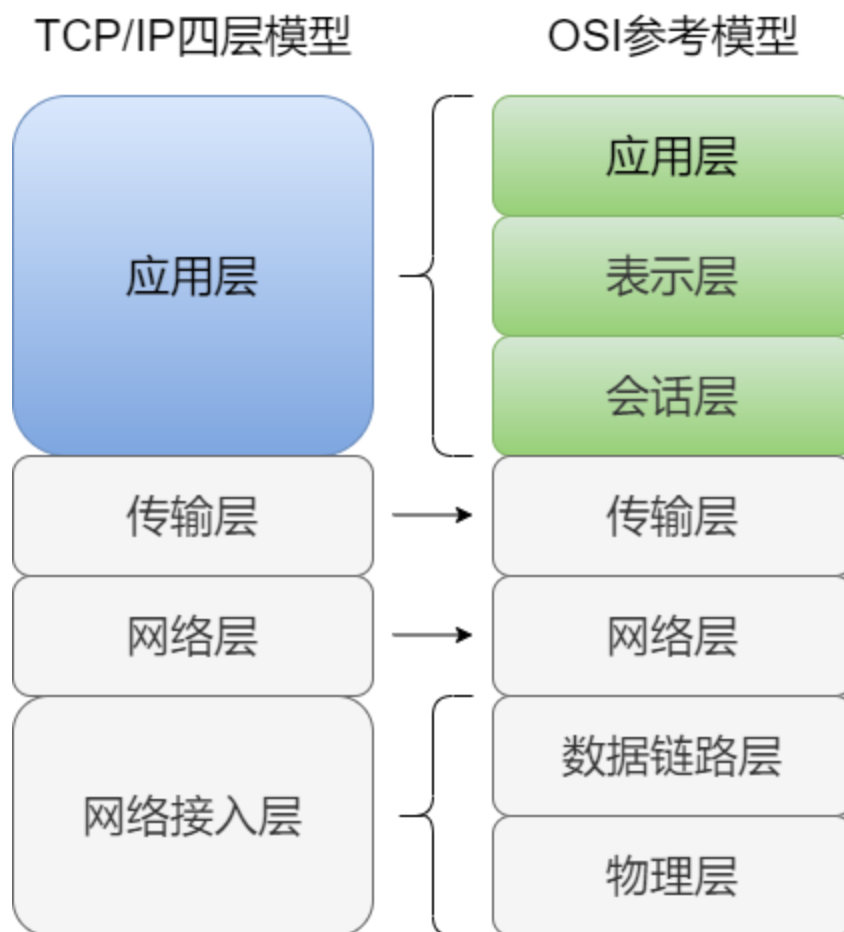


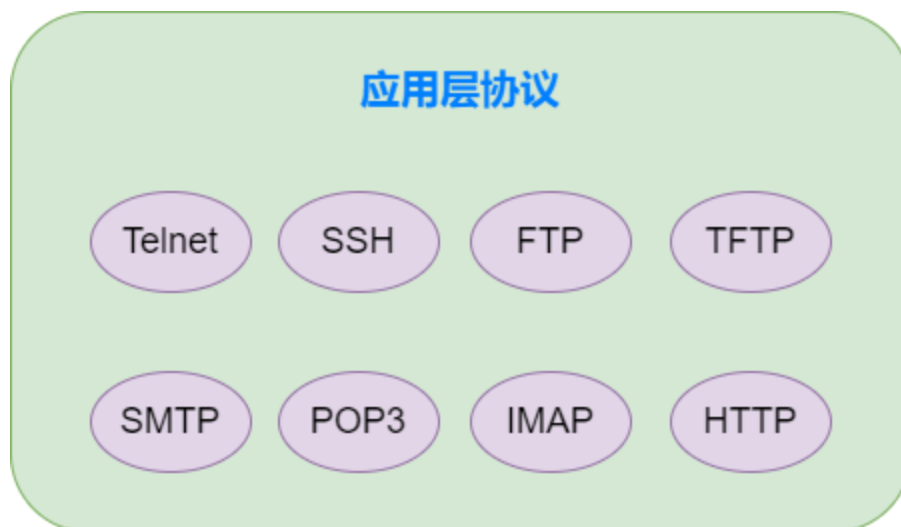
# 七 应用层

应用层为网络用户提供了丰富的应用程序功能和服务，使得用户能够进行远程数据访问、交互和共享。应用层协议是运行在网络上传输的最上层协议，也是用户需求最为直接和明显的协议。

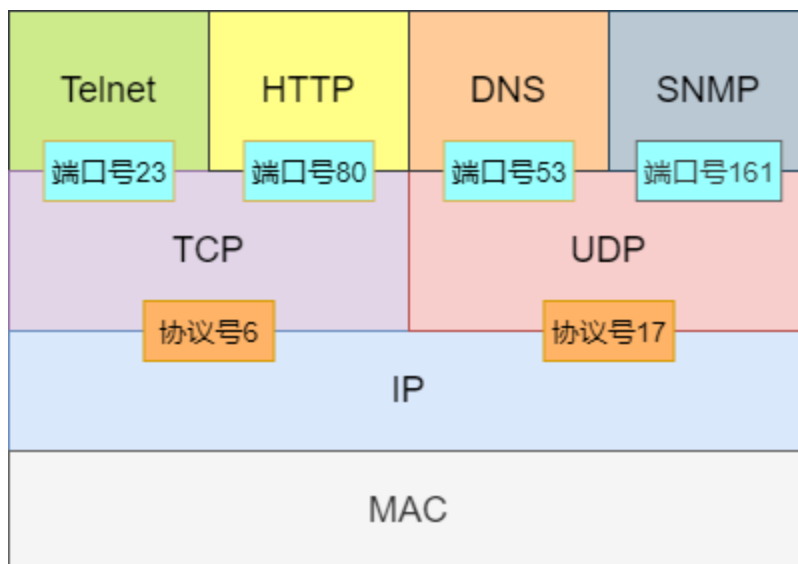
在 TCP/IP 模型中，应用层提供的服务相当于 OSI 模型的应用层、表示层和会话层的服务总和。不仅包含了**管理通信连接**的会话层功能、**数据格式转换**的表示层功能，还包括**主机间交互**的应用层功能。



**应用层的目的是向应用程序提供网络接口，直接向用户提供服务。**相比于下层的网络协议，应用协议要常见得多，可能大家都听过 HTTP、HTTPS、SSH 等应用层协议。



TCP/IP 模型中应用层位于传输层之上，传输层的端口号用于标识数据所对应的应用层协议。也就是说，有端口号的协议都是**应用层协议**。应用协议是终端设备之间的应用通信规则。应用之间交互的信息叫**消息**，应用协议定义这些消息的格式以及消息的控制或操作的规则。

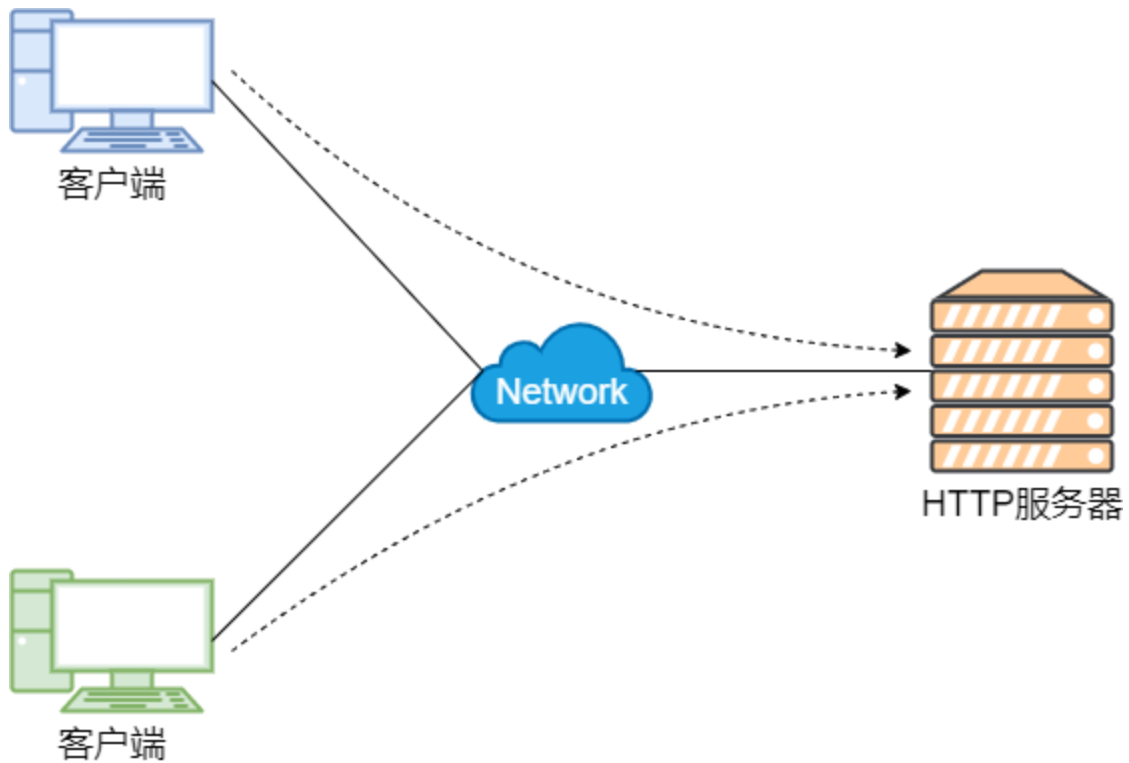


应用协议的通信方式可分为两类：

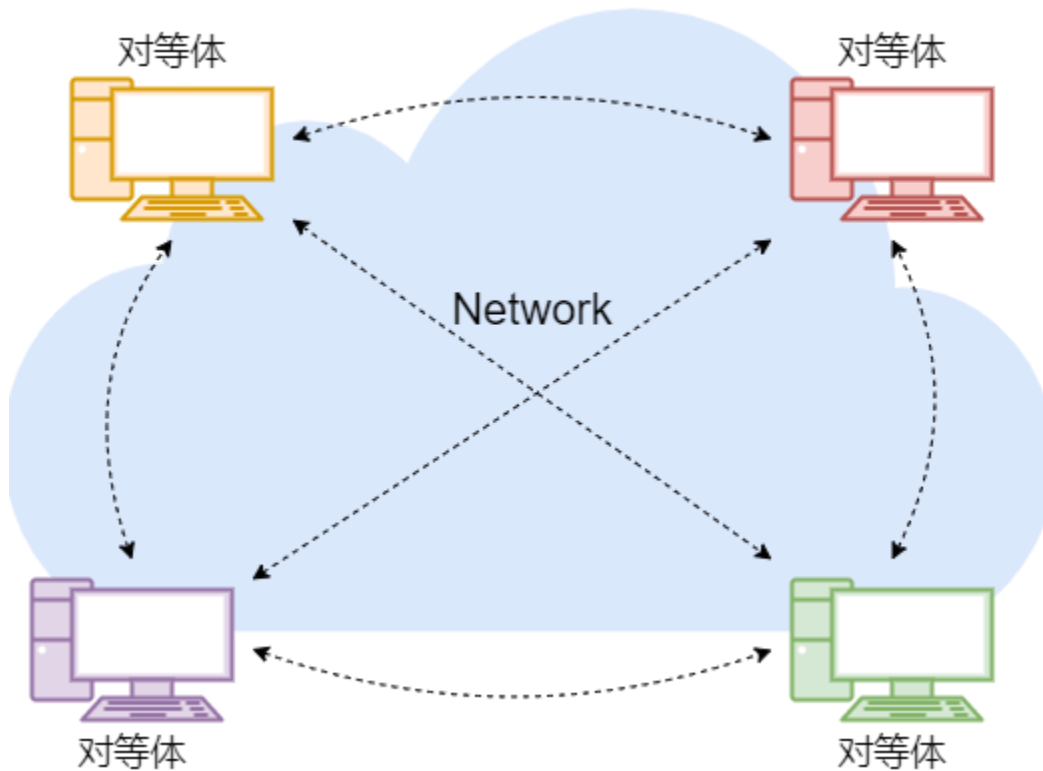
- 服务器和客户端模型
- P2P 模型

在**服务器和客户端模型**中，始终公开固定 IP 地址的主机为其它主机的应用程序提供服务，请求服务的主机之间不会互相通信。这些为其它主机提供服务的终端设备称为**服务**

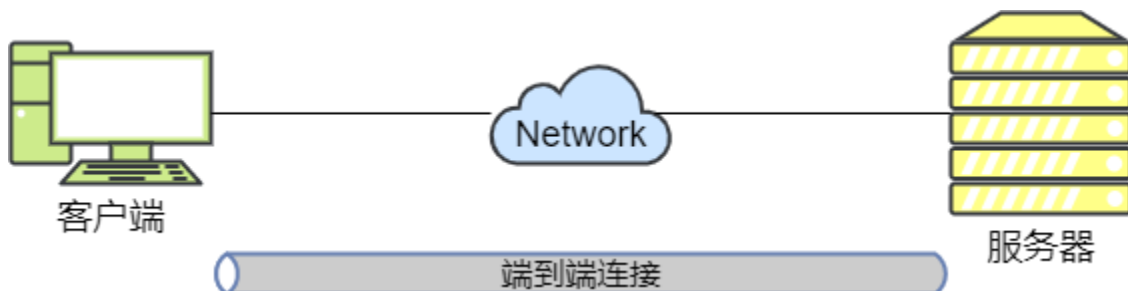
器，那些请求服务的主机则称为**客户端**。大多数应用层协议，都是这种模型。



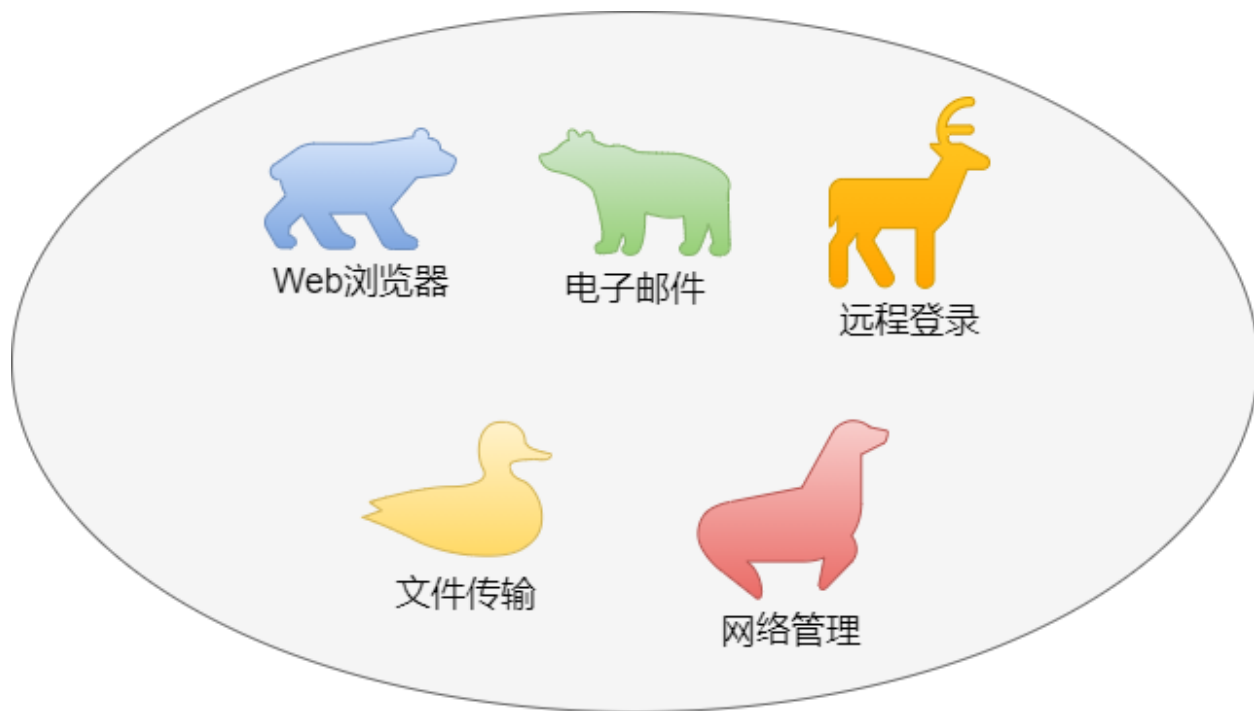
在 **P2P 模型**中，没有特定的服务器或客户端，这些设备上安装的应用程序，可以在主机间建立对等连接，既可以提供服务，也可以接受服务。通常是大流量的应用程序采用 P2P 模型，比如：下载器等。



开发**应用程序**时，为了实现相应的功能和目的，可以使用现有的应用协议，也可以自己定义一个新的应用协议。同时，应用程序可以直接使用传输层以下的网络传输服务，开发者只需要关心选择哪种应用协议、如何开发即可，而不用考虑数据是如何传输到目的地。这也是 TCP/IP 分层模型的特点。

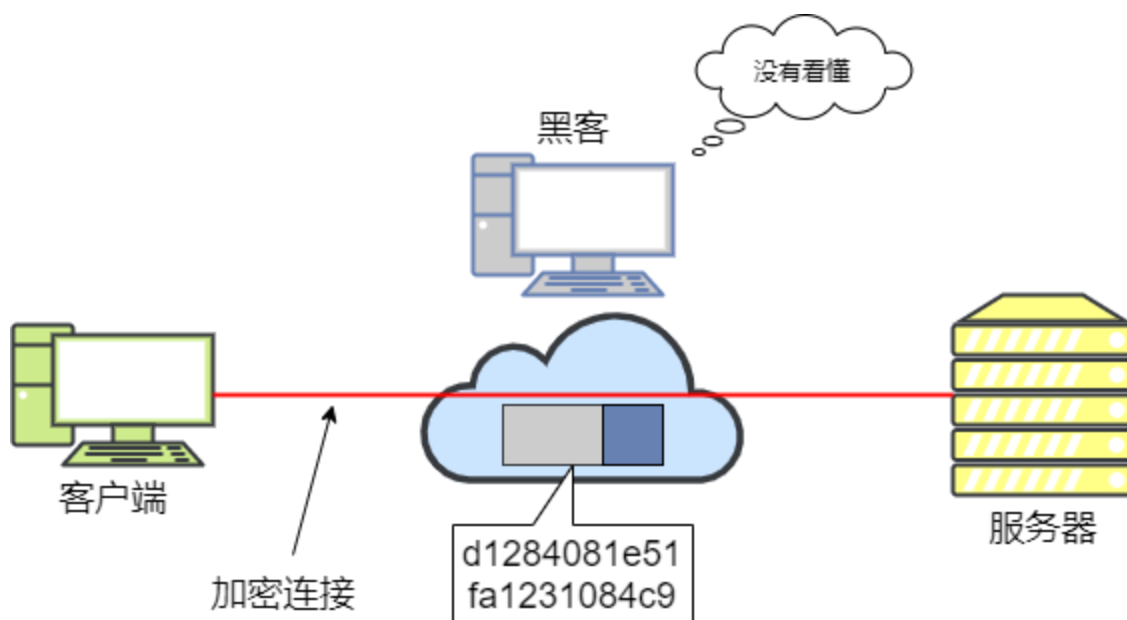


**应用程序**有很多，包括 Web 浏览器、电子邮件、远程登录、文件传输、网络管理等。这些应用程序都会使用应用协议进行通信，应用协议正是为了实现应用程序的功能而设计和创造的。



## SSH

**SSH (Secure Shell) 协议**全称是安全外壳协议，SSH 是**加密**的远程登录协议，提供更加安全的远程登录服务。使用 SSH 后会加密通信内容。即使信息被截获，由于无法解密，也无法了解数据的真正内容。



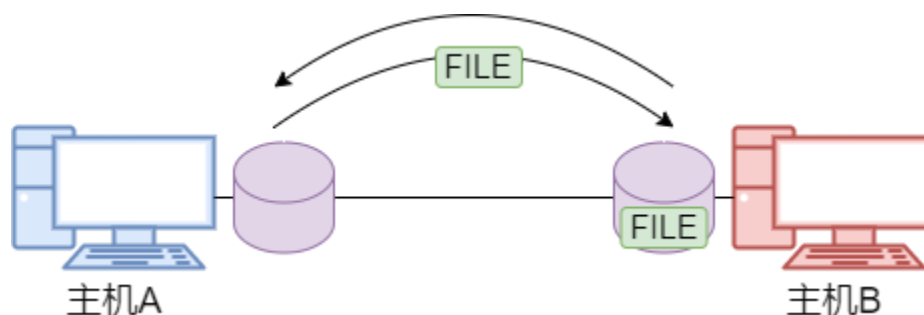
SSH 协议常用版本是 **SSHv2**，SSH 客户端通过 SSHv2 协议与 SSH 服务器建立一条 **TCP 的加密信道**，建立这条安全信道的方式是让客户端使用服务器的 RSA 公钥来验证 SSH 服务器的身份。SSH 协议默认使用 **TCP 22 端口**。

如果客户端成功验证了服务器的身份，它们之间就会创建出一个会话密钥，并用双方协商出来的加密算法和会话密钥，对这个信道传输的数据进行**加密**。这样，两台设备之间就建立了一条安全的信道，使用这条安全信道发送密码，密码以密文的形式传输，通过服务器的**身份认证**。SSH 就是通过这种方式建立加密信道，确保 SSH 服务器，也就是被管理设备的 Shell 免遭非法用户操作。

说到密钥，SSH 协议使用非对称密钥对（RSA公私钥）来认证客户端信息

## 文件传输

除了远程登录，我们还需要从远端设备传输文件，文件传输协议提供的应用服务可以满足我们的需求。**FTP** 是网络上文件传输的标准协议，FTP 使用 **TCP** 作为传输协议，支持用户的**登录认证**和**访问权限的控制**。另一种常见的文件传输协议是 **TFTP** 协议，TFTP 是一种简单的文件传输协议，不支持用户的登录认证，也没有复杂的命令。TFTP 使用 **UDP** 作为传输协议，并有重传机制。

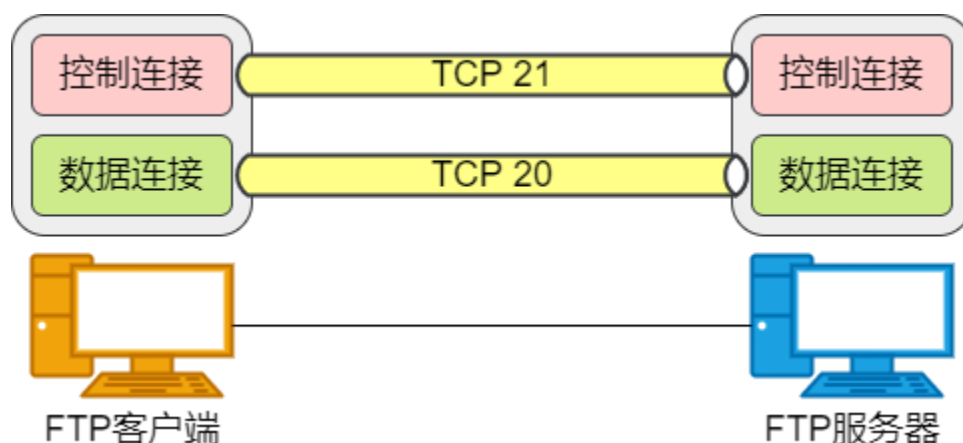


## FTP

**FTP** 用于服务器和客户端之间传输文件，是 IP 网络上传输文件的通用协议。FTP 采用客户端和服务器的模式，使用 **TCP** 协议提供可靠传输。FTP 可以对登录服务器的**用户名和密码进行验证**，允许客户端指定文件的**传输类型**，并且可以设置文件的**传输权限**。

FTP 使用两条 TCP 连接实现文件传输。一条是 **FTP 控制连接**，用来控制管理；另一条是 **FTP 数据连接**，用于数据传输。**FTP 控制连接**用于传输 FTP 控制命令和命令执行的应答信息，比如登录用户名和密码的验证、发送文件的名称、发送方式的设置。这条连接在整个 FTP 会话过程中一直保持打开，通过 ASCII 码字符串发送请求和接收应答。在控

制连接上无法发送数据，而 **FTP 数据连接** 用于文件和文件列表的传输，仅在需要传输数据时建立数据连接，数据传输完毕后终止。



FTP 控制连接使用的是 **TCP 21 号端口**，也是 FTP 服务器的侦听端口，等待客户端的连接。在 TCP 21 号端口进行文件 GET（RETR）、PUT（STOR），以及文件表（LIST）等操作时，每次都会建立一个用于数据传输的数据连接。数据和文件表的传输正式在这个数据连接上进行的。

数据连接的 TCP 连接通常使用**端口 20**。也可以使用 PORT 命令修改为其它值。

相同的一个文件，不同的操作系统可能有不同的存储方式。为了确保文件能够准确的传送给对方，常用 2 中传输模式：

- ASCII 模式

**ASCII 模式**是默认的文件传输模式。发送方把本地文件转换成标准的 ASCII 码，然后在网络中传输；接收方收到文件后，根据自己的文件存储方式，把它转换成本地文件。

ASCII 文件传输模式通常用于传输文本文件。

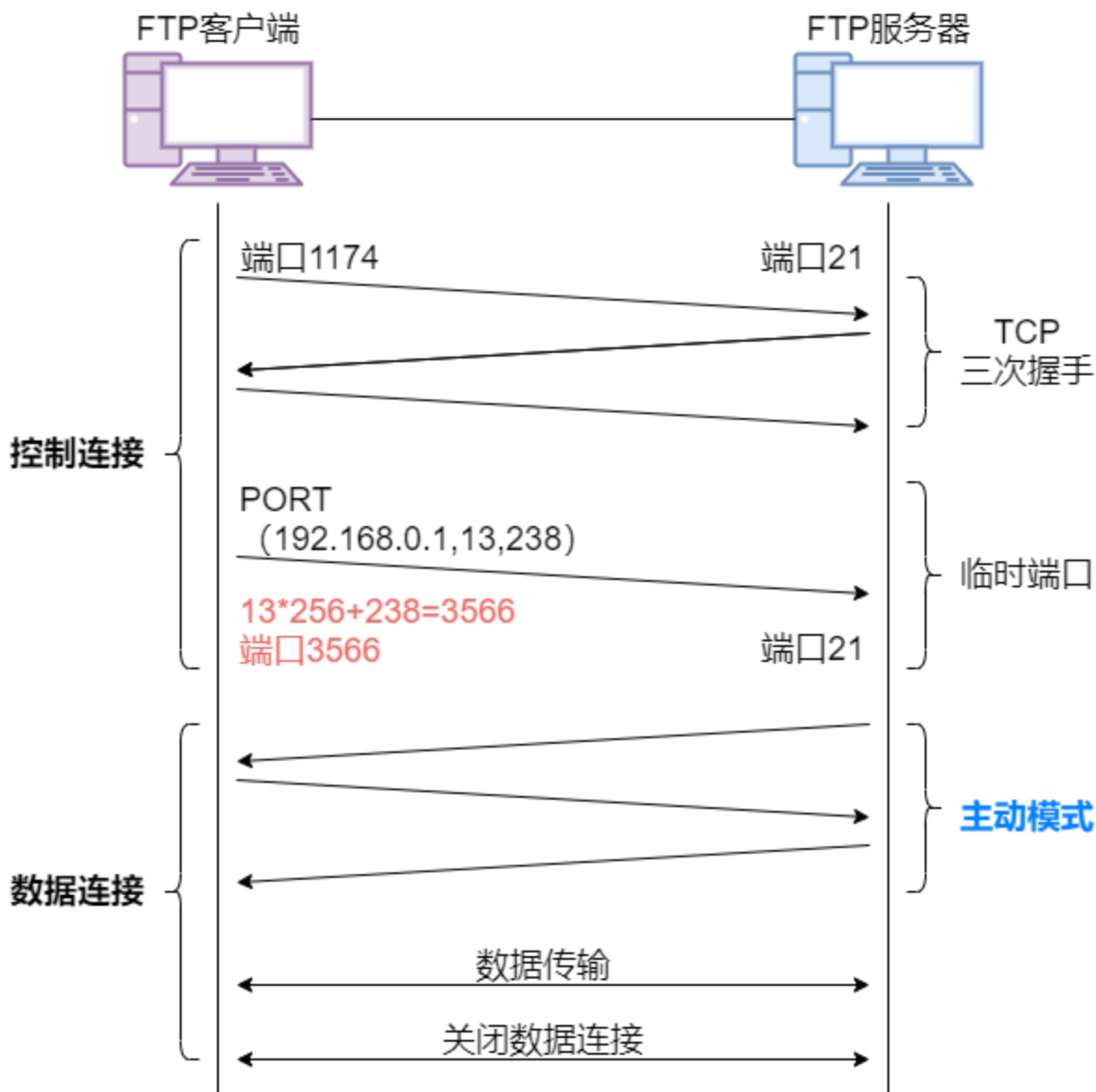
- 二进制流模式

**二进制流模式**也称为图像文件传输模式。发送方不做任何转换，把文件按照比特流的方式进行传输。二进制文件类型通常用于传送程序文件。

在 FTP 数据连接过程中，有两种数据传输方式：**主动方式**和**被动方式**。

FTP 主动传输方式，也称为 **PORT** 方式。采用主动方式建立数据连接时，FTP 客户端会通过 FTP 控制连接向 FTP 服务器发送 PORT 命令，PORT 命令携带参数：A1、A2、A3、A4、P1、P2，其中 A1、A2、A3、A4 表示需要建立数据连接的主机 IP 地址，而 P1 和 P2 表示客户端用于传输数据的临时端口号，临时端口号的数值为  $256 \times P1 + P2$

。当需要传输数据时，服务器通过 TCP 端口号 20 与客户端提供的临时端口建立数据传输通道，完成数据传输。在整个过程中，由于服务器在建立数据连接时主动发起连接，因此被称为**主动模式**。

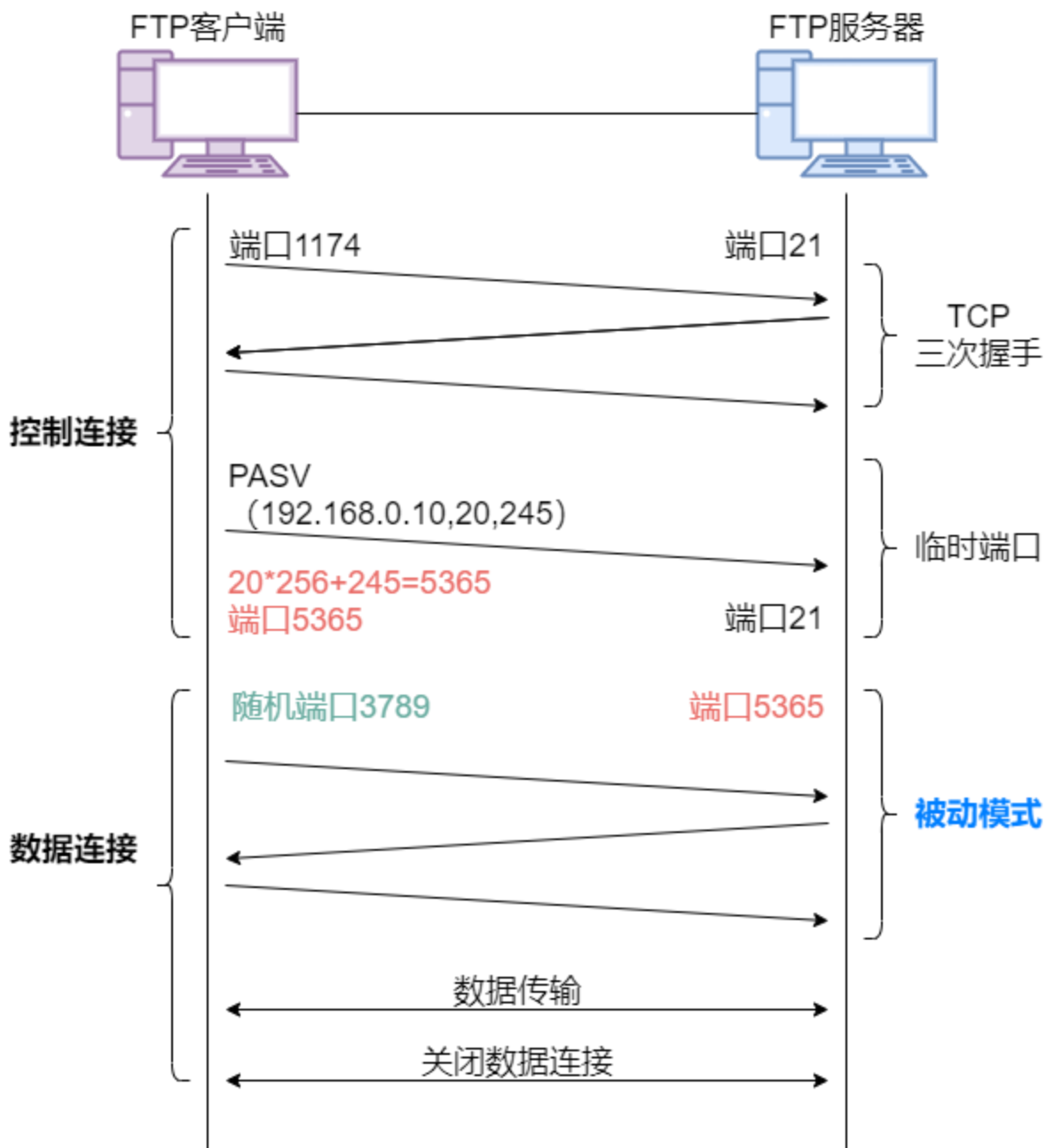


如果客户端在防火墙内部，主动方式可能会有问题，因为客户端的端口号是随机的，防火墙并不知道。默认安全策略，防火墙只会允许外部主机访问部分内部已知端口，阻断对内部随机端口的访问，从而无法建立 FTP 数据连接。这时，就需要使用 FTP 被动方式来进行文件传输。

被动方式也被称为 **PASV** 方式。FTP 控制通道建立后，希望通过被动方式建立数据传输通道的 FTP 客户端会利用控制通道向 FTP 服务器发送 PASV 命令，告诉服务器进入被



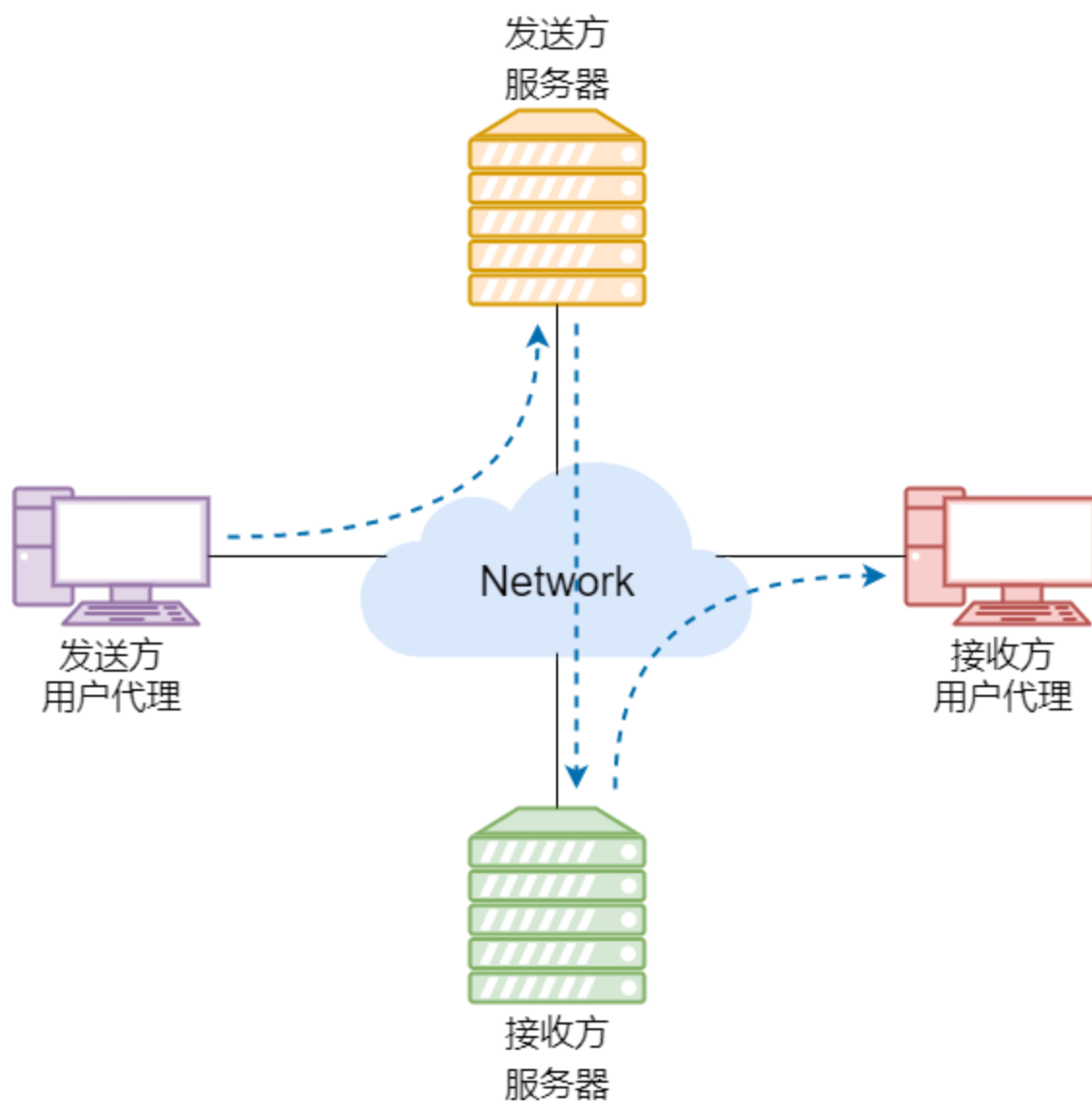
动方式传输。服务器选择临时端口号并告知客户端，命令参数和主动传输方式一致。当需要传输数据时，客户端主动与服务器的临时端口建立数据传输通道，并完成数据传输。在整个过程中，服务器是被动接收客户端的数据连接，所以被称为**被动模式**。



采用被动方式时，两个连接都由客户端发起。一般防火墙不会限制内部的客户端发起的连接，这样就解决了主动方式下的问题。

## 通信架构

电子邮件在几十年的发展过程中出现了明显的变化，从原始的发送方电脑直接向接收方电脑发送电子邮件，演变成**收发双方都使用邮件服务器代为收发邮件**。通过这种方式，电子邮件通信不再依赖接收方当前是否在线，而电子邮件的通信过程由简单的发送方到接收方，演变成**发送方电脑到发送方邮件服务器，发送方邮件服务器到接收方邮件服务器，以及接收方邮件服务器到接收方电脑**的三个通信过程。并且参与通信的四方都不是直接相连，而是分别独立连接到互联网中。这个架构中，邮件发送方和接收方使用的电脑称为**用户代理**。



## 邮件地址

使用电子邮件时，需要拥有一个地址，这个地址叫做**邮件地址**，也叫**邮箱地址**。它相当于通信地址和姓名。我的电子邮件如下：

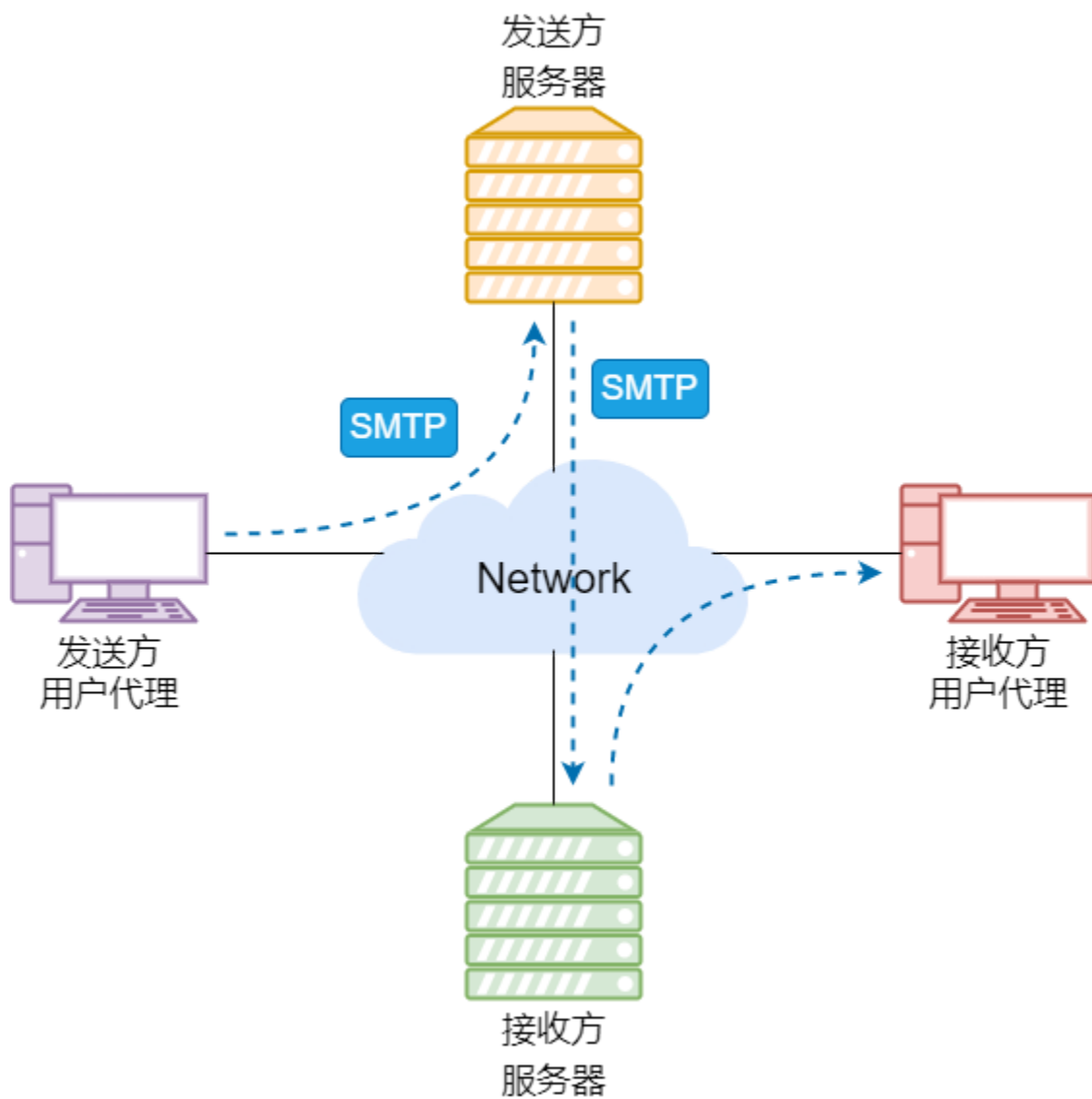
networkfox@qq.com

networkfox 表示**用户的名称**，同一个通信地址内，名称必须是唯一的，不能出现重复；@ 表示**分隔符**；<http://qq.com> 是**用户邮箱的邮件接收服务器的域名**。

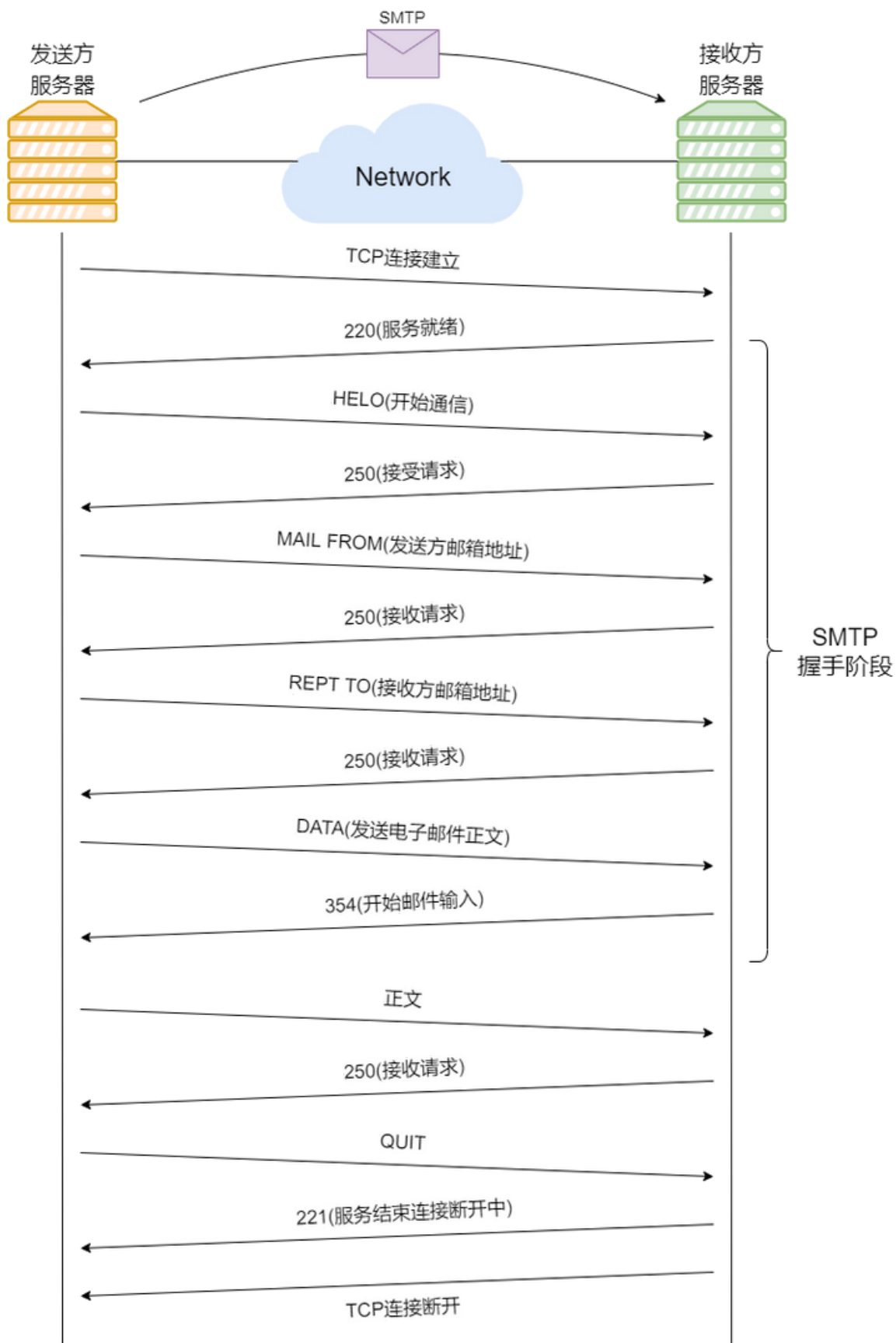
电子邮件的发送地址由 **DNS** 管理。DNS 中注册了邮件地址和对应邮件服务器的域名。这些映射信息被称为 **MX 记录**。比如：<http://qq.com> 的 MX 记录中指定了 <http://mail.qq.com>。那么任何发送给 <http://qq.com> 结尾的邮件都被发送到 <http://mail.qq.com> 服务器。就这样，根据 MX 记录中指定的邮件服务器，可以管理不同邮件地址与特定邮件服务器之间的映射关系。

## SMTP 协议

提供电子邮件服务的协议叫做 **SMTP**。SMTP 用于收发双方的邮件服务器之间，而不是用户代理和邮件服务器之间的通信方式。在实际使用中，发送方用户代理与发送方服务器之间也常采用 SMTP 协议。



SMTP 为了实现高效发送邮件内容，在传输层使用了 **TCP 协议**，**端口号是 25**。在一台邮件服务器向另一台邮件服务器发送邮件时，首先向对方的 TCP 25 端口发起一条连接。然后利用这条 TCP 连接发送控制消息和数据。



尽管 SMTP 协议的逻辑简单，也足以顺利完成邮件的传输工作，但难免存在一些安全缺陷：

- SMTP 传输的邮件是**明文**的形式，没有提供数据加密机制，可以看到邮件传输的具体内容，用户信息的机密性无法得到保障。
- SMTP 没有提供任何**认证机制**，即使使用了伪造的发件人邮件地址也无法识别，会出现冒名顶替的安全问题。

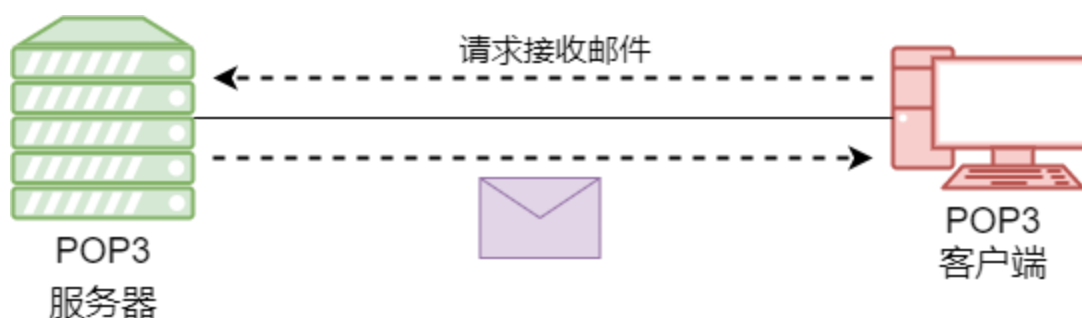
漫天的广告邮件和包含钓鱼链接的垃圾邮件成为日益严重的问题。为了修正 SMTP 出现的问题，IETF 定义了扩展的 SMTP，即 ESMTP。**ESMTP** 提供的扩展功能中包括**认证机制**和**加密机制**等。

在整个邮件传输的过程中，SMTP/ESMTP 协议定义了邮件服务器之间的消息传输方式。在接收服务器收到电子邮件后，接收方（用户代理）是如何访问邮件则需要其它的协议来处理。

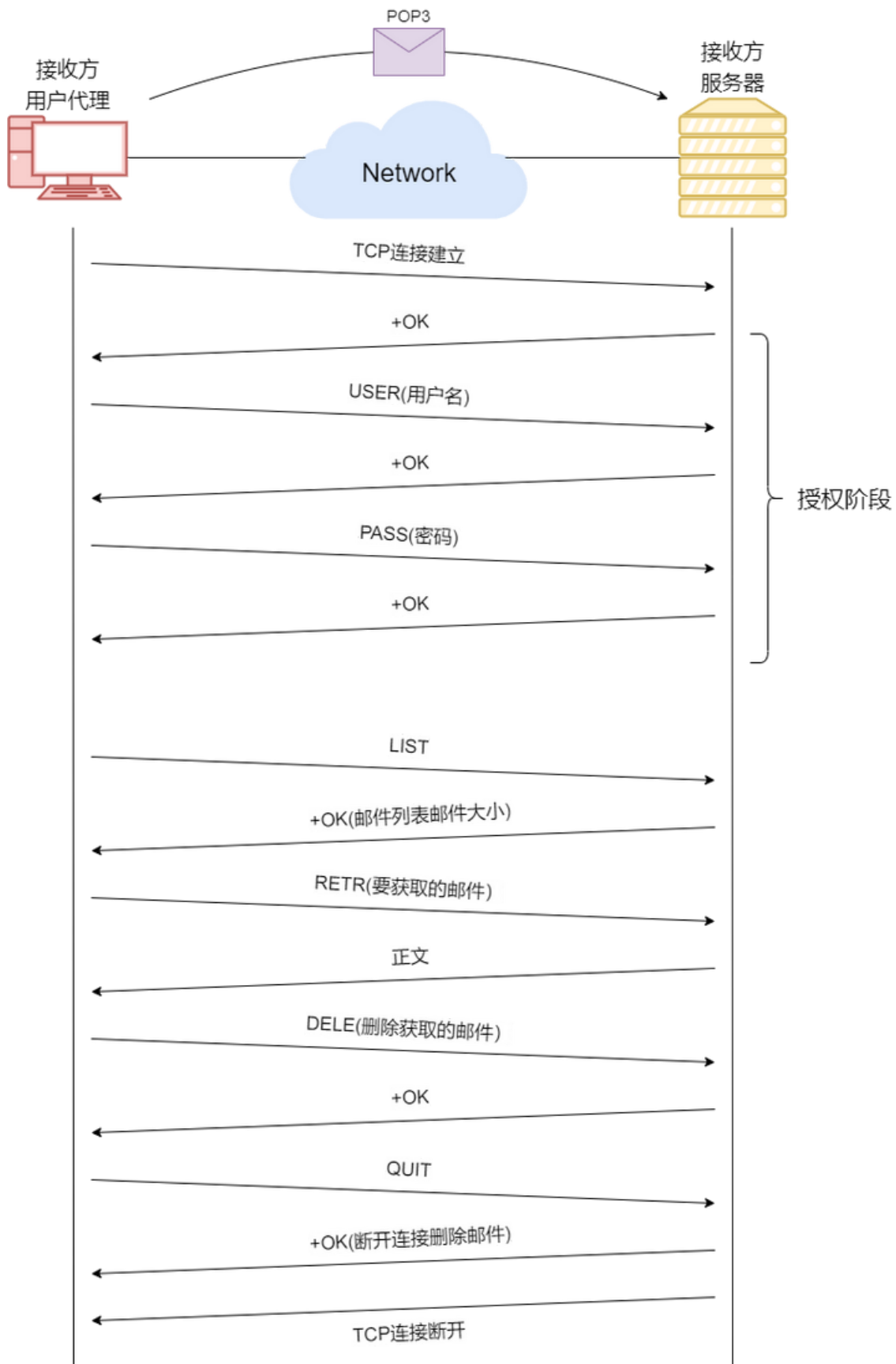
## POP3 协议

电子邮件通过 SMTP 协议到达接收方服务器，个人电脑不可能长期处于开机状态，用户希望一开机就能收到邮件，然而 SMTP 没有这种功能。

为了解决这个问题，就引入了 POP3 协议。**POP3** 协议是用于接收电子邮件的协议。发送端的邮件使用 SMTP 协议将电子邮件转发给一直在线的 POP3 服务器。客户端再根据 POP3 协议从 POP3 服务器接收邮件。这个过程中，为了防止别人盗取邮件内容，还要进行用户认证。



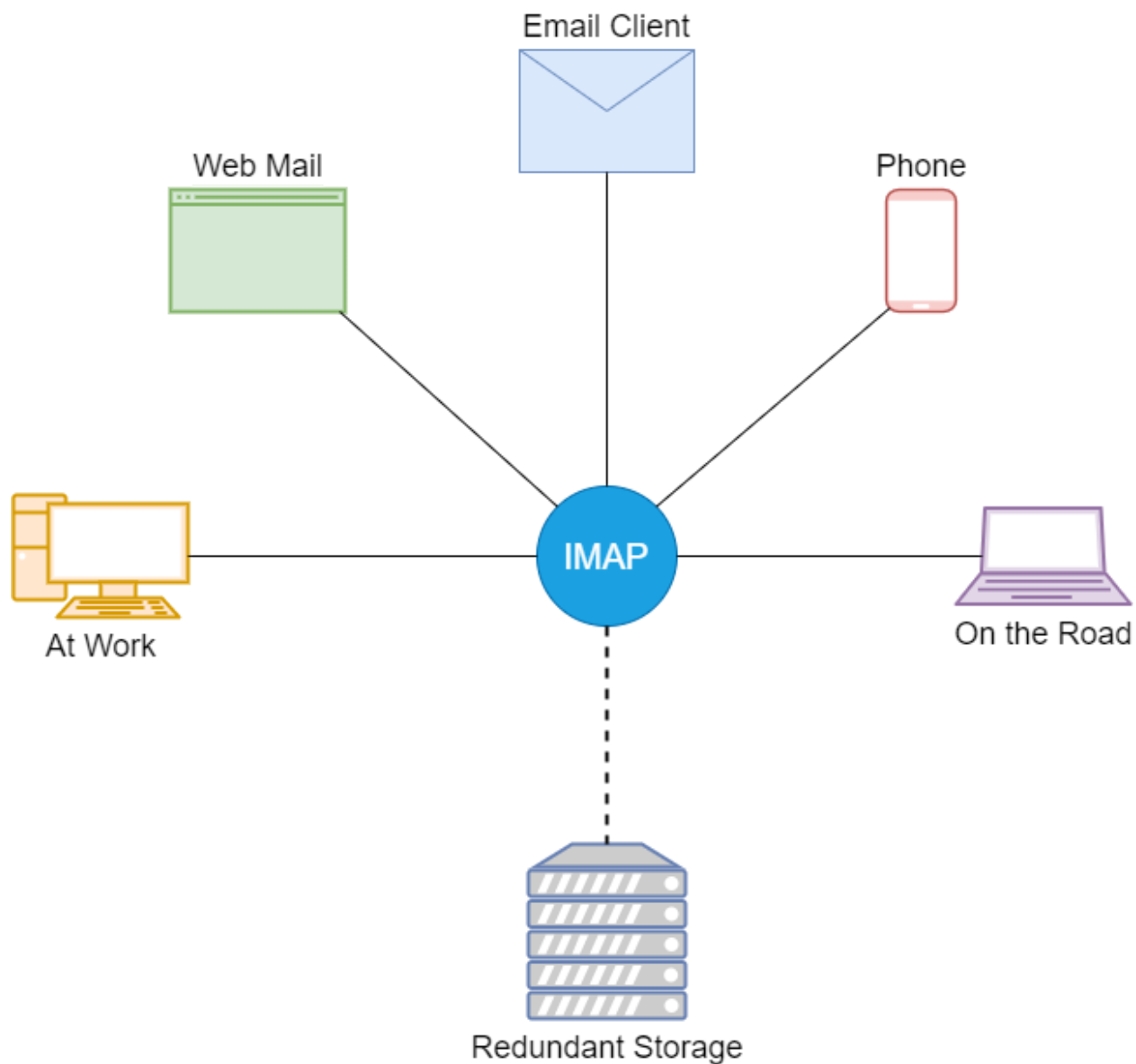
POP3 协议和 SMTP 协议一样，是基于 TCP 的应用层协议，使用 **TCP 110 端口** 连接邮件服务器。接收方的邮件客户端程序首先使用 TCP 连接到 POP3 服务器的 TCP 端口 110；再进行**用户认证**、**邮件列表查询**、**邮件下载**、**邮件删除**等操作；操作完成后，客户端与邮件服务器之间再断开 TCP 连接。



POP3 仅负责邮件的下载，邮件从客户端上传到邮件服务器由 SMTP 协议完成。

## IMAP 协议

POP3 协议的邮件客户端能够在邮件服务器上执行的操作很少，而且邮件要下载到客户端本地，而不保留在邮件服务器，实际使用时很不方便。目前使用更广泛的接收电子邮件的协议是 **IMAP**。在 IMAP 中邮件则由服务器进行管理。



使用 IMAP 时，不必从服务器上下载所有的邮件也可以查看。由于 IMAP 是在服务器端处理 MIME 信息，它可以实现邮件附件的选择性下载功能。比如：一封邮件有 5 个附件

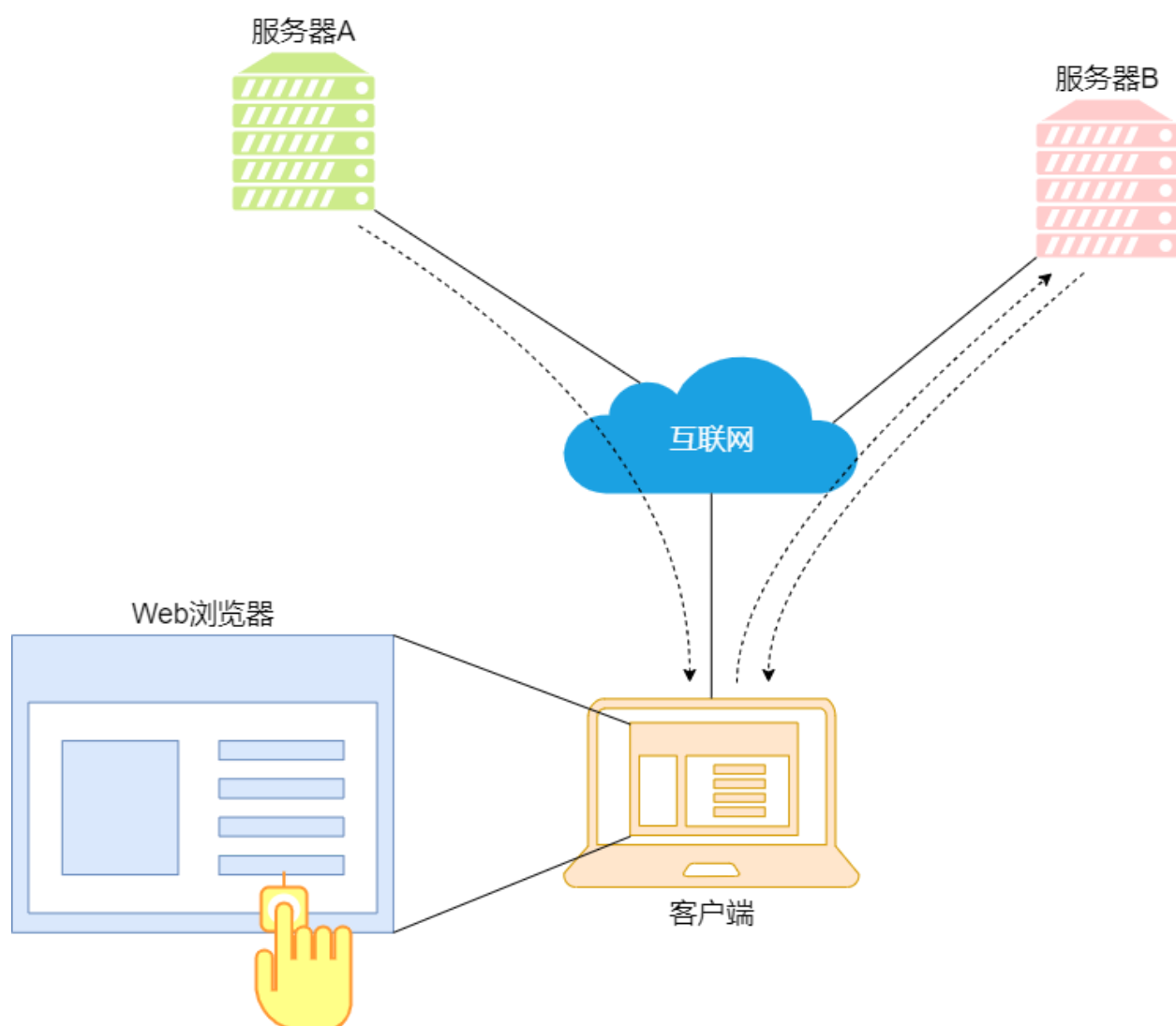


时，可以只下载其中的 3 个附件。IMAP 还会在服务器上对“已读/未读”信息和邮件分类进行管理，所以在不同的电脑上打开邮箱，也能保持同步，使用起来非常方便。

## WWW

**万维网（www）**是将互联网的信息以超文本形式展现的系统，也叫做**Web**。可以显示 WWW 信息的客户端软件叫做**Web 浏览器**，有时简称为浏览器。目前常用的 Web 浏览器包括微软的 Internet Explorer、谷歌公司的 Google Chrome、Firefox 以及 Apple 公司的 Safari 等。

使用浏览器，我们不需要关心信息保存在哪个服务器，只需轻轻点击鼠标，就可以访问页面上的链接并打开相关信息。



通过浏览器进行访问后，显示在浏览器上的内容叫做 **Web 页**。访问一个网站时看到的一个页面称为**首页**（又称为主页）。很多公司的主页地址形式如下：

http://www.公司名称.com.cn

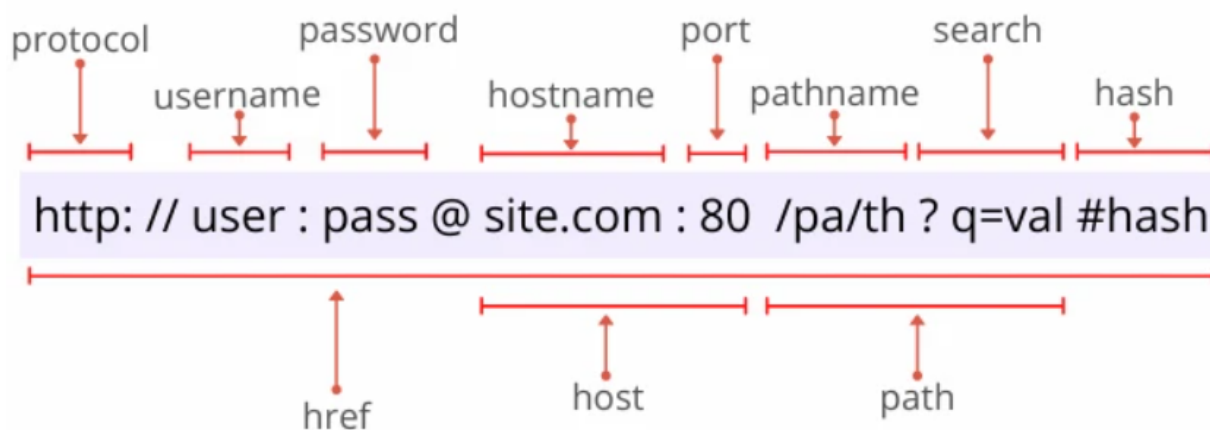
这类主页中通常有公司概况、产品信息、招聘信息等内容。我们可以点击这些标题的图标或链接，就可以跳转到对应的页面上。这些页面上的信息不仅仅是文字内容，还有图片或动画，甚至是声音或其它程序等各种各样的信息。我们不但可以通过 Web 页获取信息，还可以自己制作 Web 页向全世界发布信息。

WWW 有 3 个重要的概念，它们分别是访问信息的方式和位置（**URI**）、信息的表现形式（**HTML**）以及信息传输（**HTTP**）等操作。

## URL 与 URI

- URI是用来标识资源的字符串文本（Uniform Resource Identifier）
- URL代表着是统一资源定位符（UniformResourceLocator）

简单来说，URI 就是由某个协议方案表示的定位资源的标识符。是将一个资源与另一个资源区分开来的字符序列。我们熟悉的 URL 地址，例如 `http://www.yaohaixiao.com`。采用 HTTP 的时候，“协议方案”就是 HTTP。URI 支持的协议方案有很多（ftp、mailto、file、news 等）。



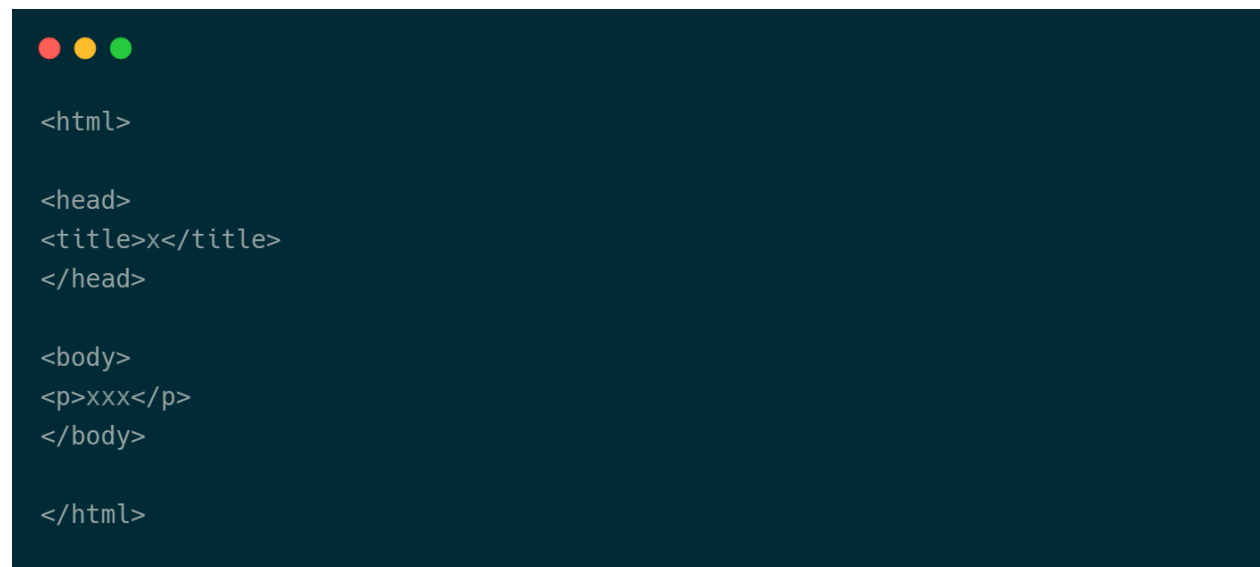
URL是一种特定类型的URI，是互联网上的资源的地址。它用于指定常见的互联网上的资源（如网页、图像、视频、页面框架、CSS样式表、代码包等）的位置。在URL中，我

们需要指定一个具体的协议，例如 HTTP、FTP 或 HTTPS 等，以告诉浏览器如何访问资源。

因此，URL和URI的主要区别在于URL是一种用来指定互联网上资源位置的特定类型的URI

## HTML

**HTML** 是用来描述 Web 页的一种语言。它可以指定浏览器中显示的文字、文字的大小和颜色，还可以对图像、动画或音频进行设置。

A dark-themed code editor window with three colored window control buttons (red, yellow, green) in the top-left corner. The editor contains the following HTML code:

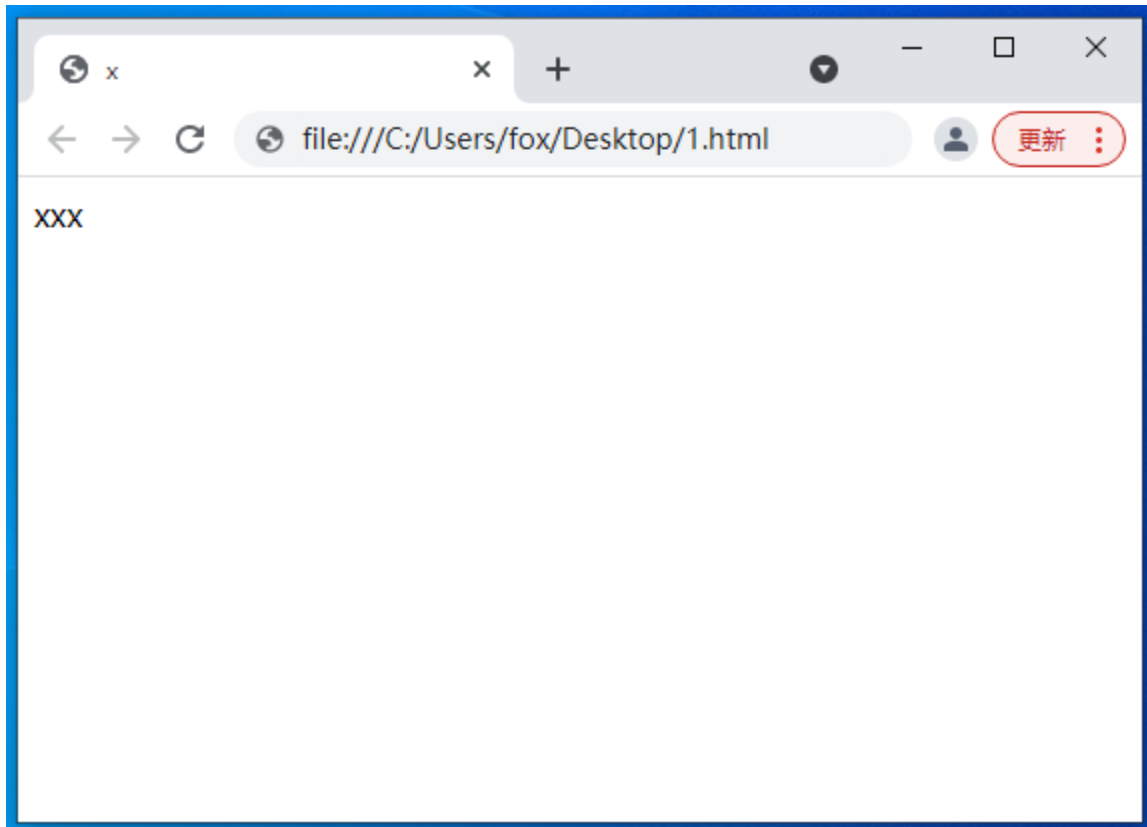
```
<html>

<head>
<title>x</title>
</head>

<body>
<p>xxx</p>
</body>

</html>
```

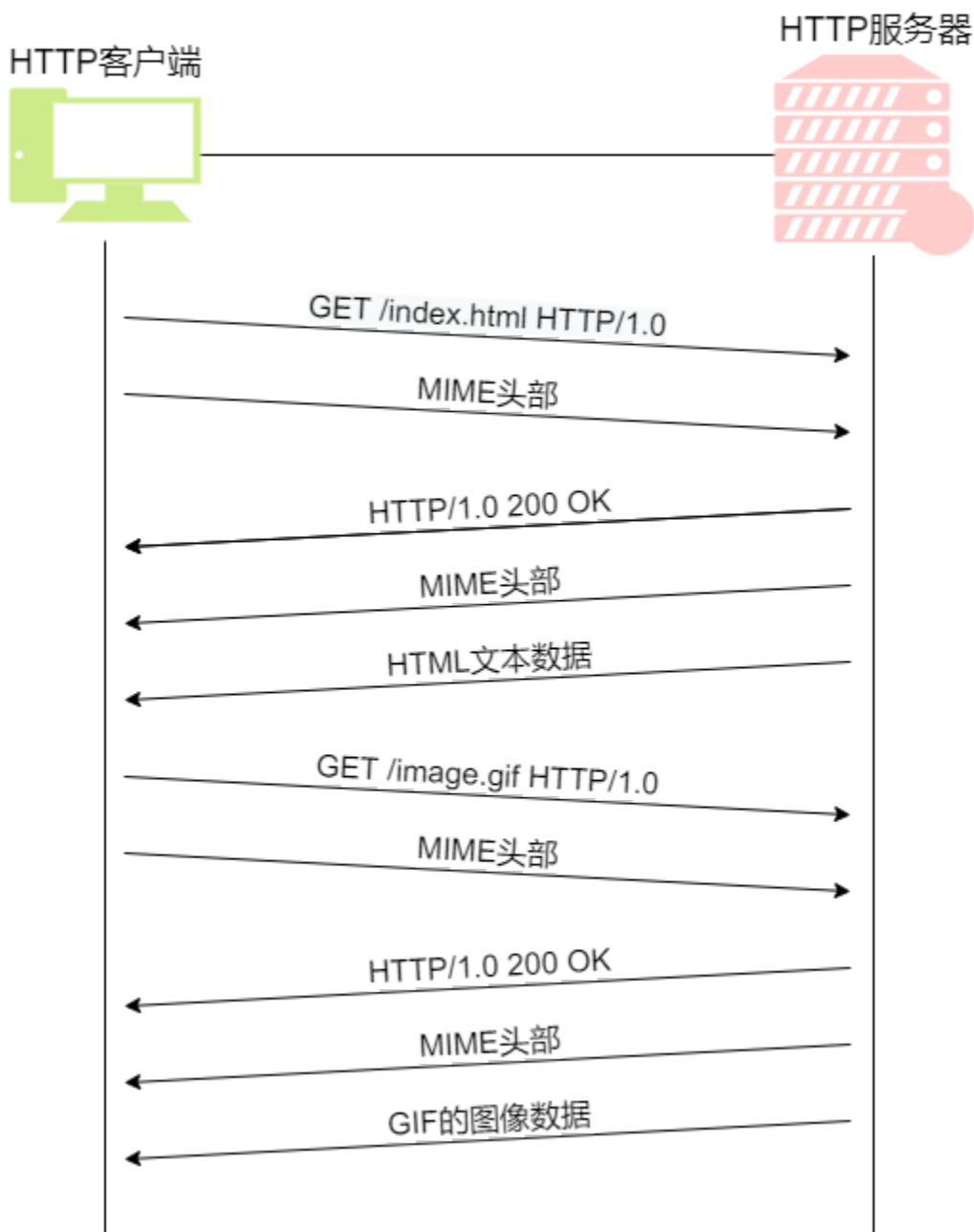
在页面中 HTML 不仅可以文字或图片附加**链接**，点击链接时还可以呈现链接所指的内容。互联网中任何一个 WWW 服务器中的信息都可以以链接的方式展现。



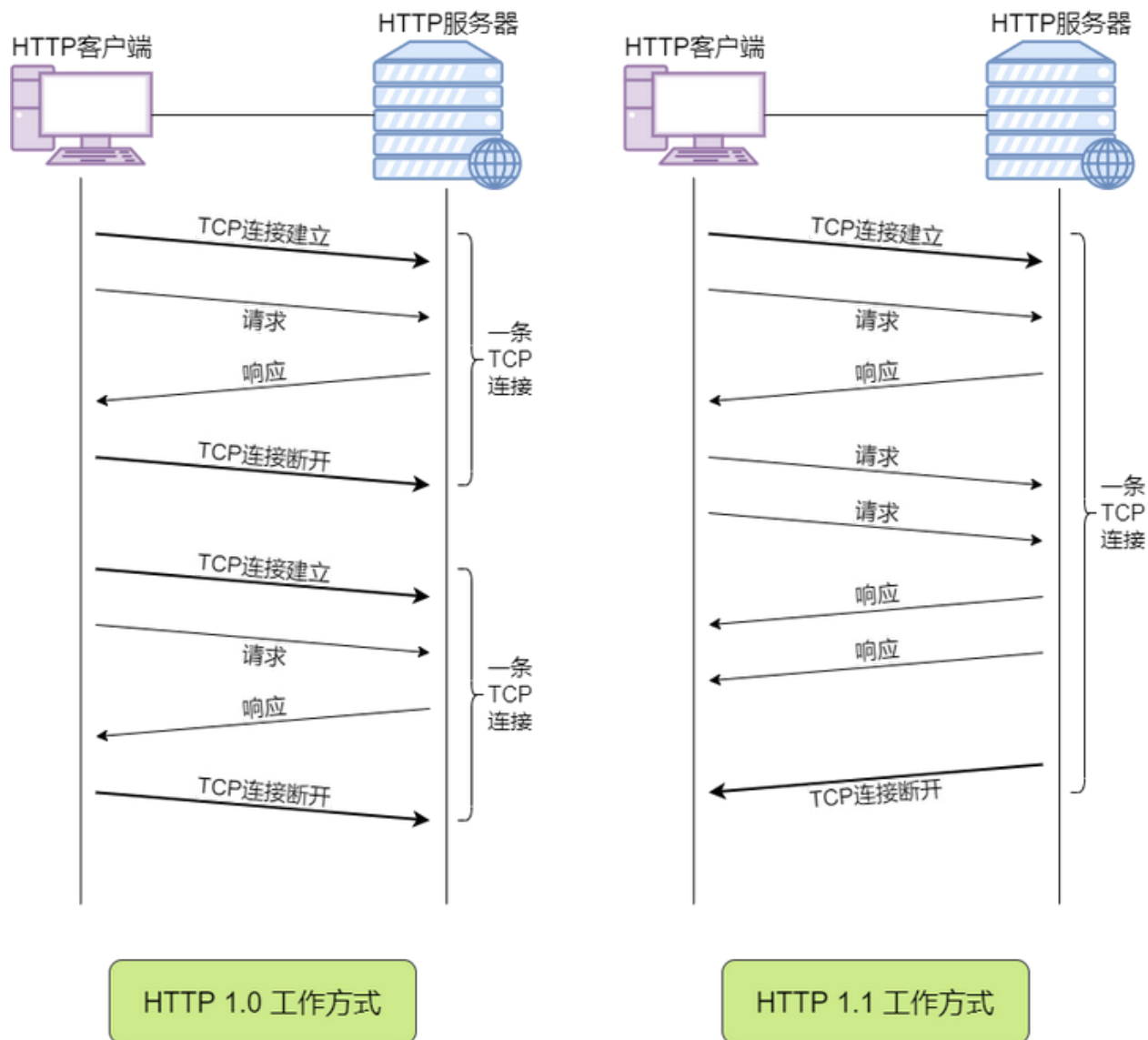
**HTML 也可以说是 WWW 的数据表现协议。**只要是用 HTML 展现的数据，即使是在不同的计算机上，效果基本上是一样的。

## HTTP

当用户在浏览器的地址栏里输入 Web 页的 URL 后，HTTP 的处理就开始了。**HTTP** 默认使用 80 端口。它的工作机制，首先是客户端向服务器的 80 端口建立一个 TCP 连接，然后在这个 TCP 连接上进行请求和应答以及数据报文的发送。



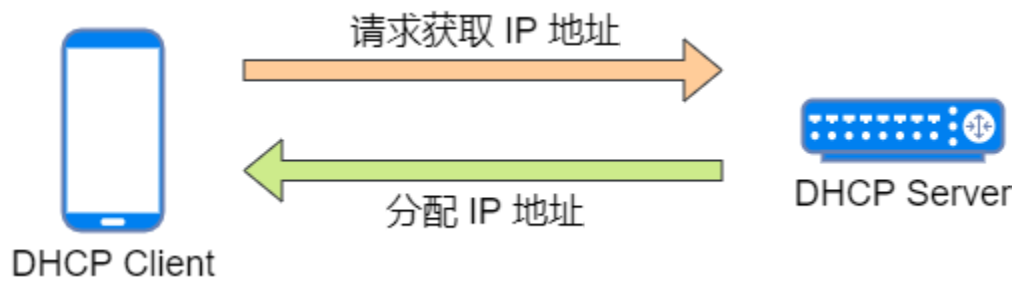
HTTP 中常用的有两个版本，**HTTP 1.0**、**HTTP 1.1**。在HTTP 1.0 中每一个命令和应答都会触发一次 TCP 连接的建立和断开。而从 HTTP 1.1 开始，允许在一个 TCP 连接上发送多个命令和应答，这种方式也叫**保持连接**（keep-alive）。可以大量减少 TCP 连接的建立和断开操作，提高传输效率。HTTP/2是在 HTTP/1.1 的基础上进行了一系列改进，如引入了二进制格式、多路复用等新特性，从而更加高效地处理数据传输。



## 网络管理应用

很多应用层协议广为人知，是因为我们在日常上网的过程中，会大量使用与这些应用协议有关的应用程序，这类应用协议称为**终端用户应用协议**；另外还有一些应用协议在网络中广泛使用，但我们对它们却少有听闻，最多在网络无法正常使用时，才会意识到它们的存在，这类应用层协议称为**系统应用协议**。

在日常工作中，网络工程师经常使用到的系统应用协议有 DHCP 协议和 DNS 协议。



详细内容可以查看往期文章《[37 张图详解 DHCP：给你 IP 地址的隐形人](#)》和《[36 张图详解 DNS：网络世界的导航](#)》。

