单工，半双工，全双工

一、单工

1、数据只在一个方向上传输，不能实现双方通信。

2、例如：电视、广播。

二、半双工

1、允许数据在两个方向上传输，但是同一时间数据只能在一个方向上传输，其实际上是切换的单工。

2、例如：对讲机。

三、全双工

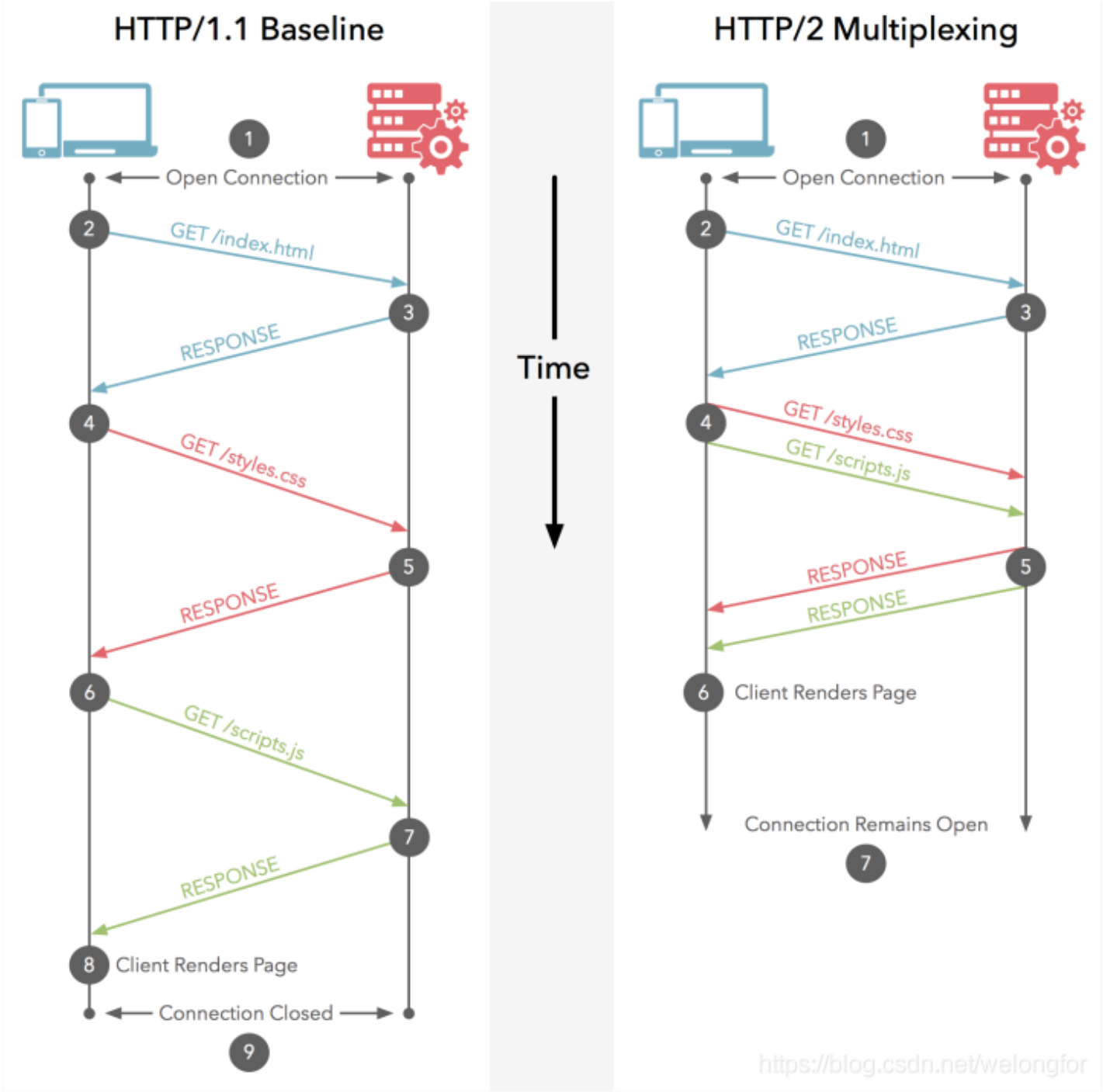
1、允许数据在两个方向上同时传输。

2、例如：手机通话，websocket

http 协议是全双工还是半双工？

分版本，版本不同，工作模式不同

1. http1.0：单工。因为是短连接，客户端发起请求之后，服务端处理完请求并收到客户端的响应后即断开连接。
2. http1.1：半双工。默认开启长连接keep-alive（在不同服务器上时间不一样，可以设置），开启一个连接可发送多个请求。
3. http2.0：全双工，允许服务端主动向客户端发送数据。



webSocket（协议）和Socket（接口）的概念介绍

一、概念

1、socket翻译为套接字，socket是在应用层和传输层之间的一个抽象层，它把TCP/IP层复杂的操作抽象为几个简单的接口供应用层调用以实现进程在网络中通信。

2、WebSocket协议是基于TCP的一种新的网络协议，和http协议一样属于应用层协议，是一种让客户端和服务器之间能进行双向实时通信的技术。

二、webSocket和Socket的区别

1、原理上的区别：

Socket是传输控制层的接口，WebSocket是应用层协议。

Socket是应用层与TCP/IP协议族通信的中间软件抽象层，它是一组接口（不是协议，为了方便使用TCP或UDP而抽象出来的一层，是位于应用层和传输控制层之间的一组接口）。
在设计模式中，Socket其实就是一个门面模式，它把复杂的TCP/IP协议族隐藏在Socket接口后面。利用TCP/IP协议建立TCP连接。（TCP连接则更依靠于底层的IP协议，IP协议的连接则依赖于链路层等更低层次。）
WebSocket则是一个典型的应用层协议。

2、灵活运用的程度不同：

WebSocket 更易用，而 Socket 更灵活。Socket是应用层与TCP/IP协议族通信的中间软件抽象层，它是一组接口。
在设计模式中，Socket其实就是一个门面模式，它把复杂的TCP/IP协议族隐藏在Socket接口后面，对用户来说，一组简单的接口就是全部，让Socket去组织数据，以符合指定的协议。
主机 A 的应用程序要能和主机 B 的应用程序通信，必须通过 Socket 建立连接，而建立 Socket 连接必须需要底层 TCP/IP 协议来建立 TCP 连接。建立 TCP 连接需要底层 IP 协议来寻址网络中的主机。
网络层使用的 IP 协议可以帮助我们根据 IP 地址来找到目标主机，但是一台主机上可能运行着多个应用程序，如何才能与指定的应用程序通信就要通过 TCP 或 UPD 的地址也就是端口号来指定。这样就可以通过一个 Socket 实例唯一代表一个主机上的一个应用程序的通信链路了。
而 WebSocket 则不同，它是一个完整的 应用层协议，包含一套标准的 API 。

3、传输层次不同：

Socket 是传输控制层的接口。用户可以通过 Socket 来操作底层 TCP/IP 协议族通信。
网络中的 Socket 并不是什么协议，而是为了使用 TCP，UDP 而抽象出来的一层 API，它是位于应用层和传输层之间的一个抽象层。Socket 是对 TCP/IP 的封装；HTTP 是轿车，提供了封装或者显示数据的具体形式；Socket 是发动机，提供了网络通信的能力。
在 Unix 一切皆文件哲学的思想下，Socket 是一种"打开—读/写—关闭"模式的实现，服务器和客户端各自维护一个"文件"，在建立连接打开后，可以向自己文件写入内容供对方读取或者读取对方内容，通讯结束时关闭文件。
WebSocket 是一种在单个 TCP 连接上进行全双工通信的协议。WebSocket 使得客户端和服务器之间的数据交换变得更加简单，允许服务端主动向客户端推送数据。
在 WebSocket API 中，浏览器和服务器只需要完成一次 HTTP 握手，两者之间就直接可以创建持久性的连接，并进行双向数据传输。

为什么有了 HTTP 还需要 WebSocket？

一般情况下我们使用 HTTP 有一个很大的缺陷，就是 HTTP 只能由客户端来主动发起，如果有需要服务端主动通知的业务，就需要轮训。轮询的效率低，非常浪费资源。为了解决 Web 端即时通讯的需求就出现了 WebSocket。