QUECTEL

Build a smarter world

移远通信

嵌入式软件开发课程



软件部-蒙政威

01 WebSoket协议简介

Part One

Socket概述

Socket（套接字）是操作系统提供给应用程序的网络编程接口。应用程序不需要直接操作 TCP/IP 协议栈，只要"插进"这个插座，就能在两台主机之间收发字节流。一个 Socket 由 **IP 地址 + 端口号 + 协议** 三元组唯一确定。

服务器 Server

1. socket() 创建套接字
2. bind() 绑定 IP:Port
3. listen() 开始监听
4. accept() 等待连接
5. read/write 收发数据
6. close() 关闭连接

TCP 字节流通道

IP:Port ↔ IP:Port

客户端 Client

1. socket() 创建套接字
2. connect() 发起连接

↓ 三次握手 ↓

3. write/read 收发数据
4. close() 关闭连接

IP 地址

标识网络上的主机

端口号

标识主机上的进程

协议

TCP / UDP

WebSocket协议

WebSocket 是 HTML5 开始提供的、在单个 TCP 连接上进行全双工（Full-Duplex）通信的应用层协议。它在 HTTP 握手阶段通过 **Upgrade: websocket** 头将 HTTP 协议升级为 WebSocket 协议，此后客户端与服务器之间可互相、随时地发送数据，不再是“请求-响应” 模式。
`Web` 代表：适配 Web 场景（浏览器），握手阶段走 HTTP，升级成双向长连接；
`Socket` 是套接字，表示“双向通信” 这个概念。

01 全双工通信
建连后客户端与服务器可互相主动推送数据，无需等待对方请求，真正的双向通道。

02 轻量帧头
协议头仅 2-10 字节（含 masking key），相比 HTTP 头部 500+ 字节大幅减少冗余。

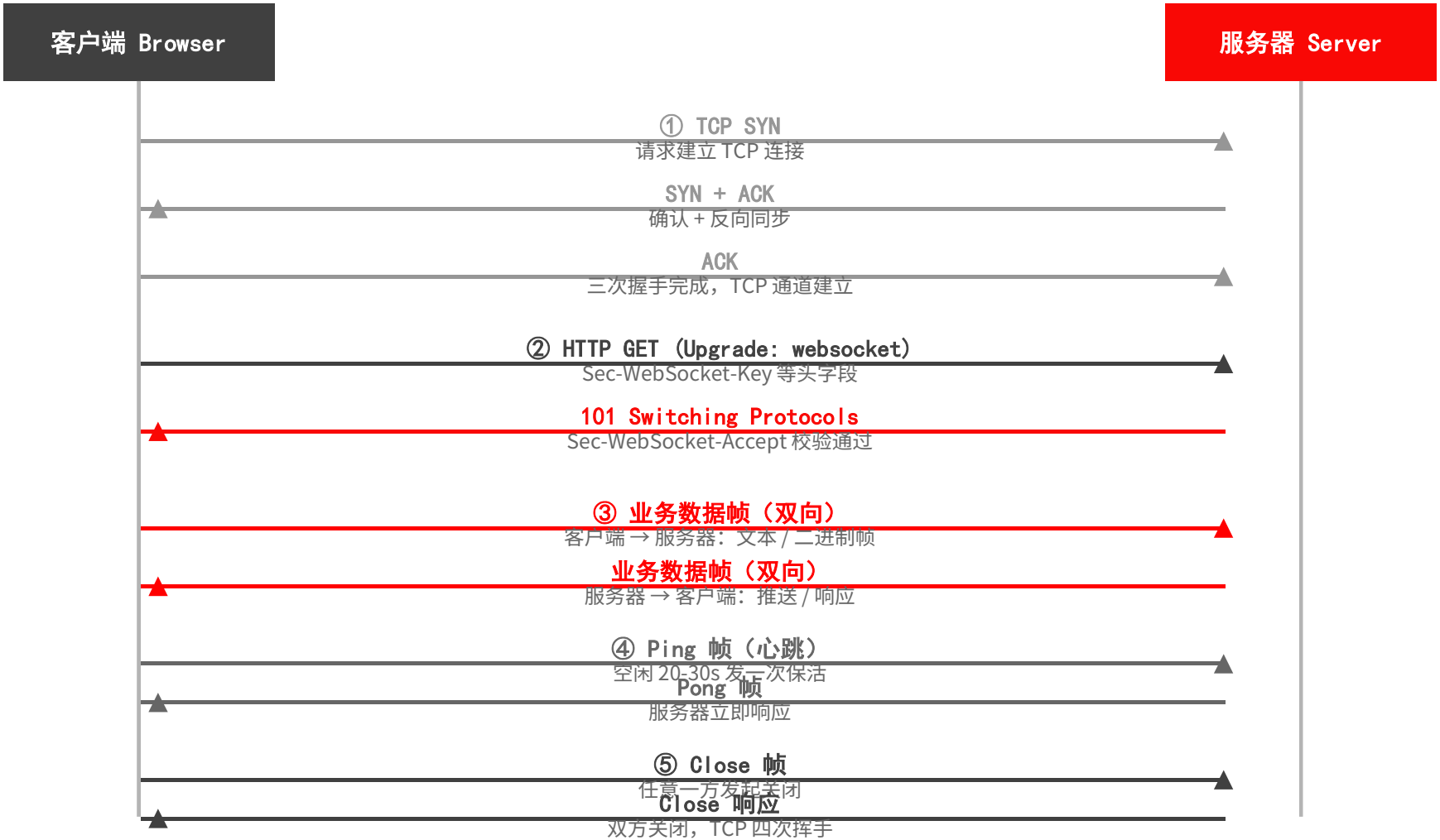
03 基于 HTTP 握手
复用 80/443 端口，通过 Upgrade: websocket 字段升级协议，穿透防火墙与代理友好。

04 持久连接
一次建连长期保持，避免反复 TCP/TLS 握手，适合高频、小尺寸的实时消息场景。

02 WebSocket工作流程

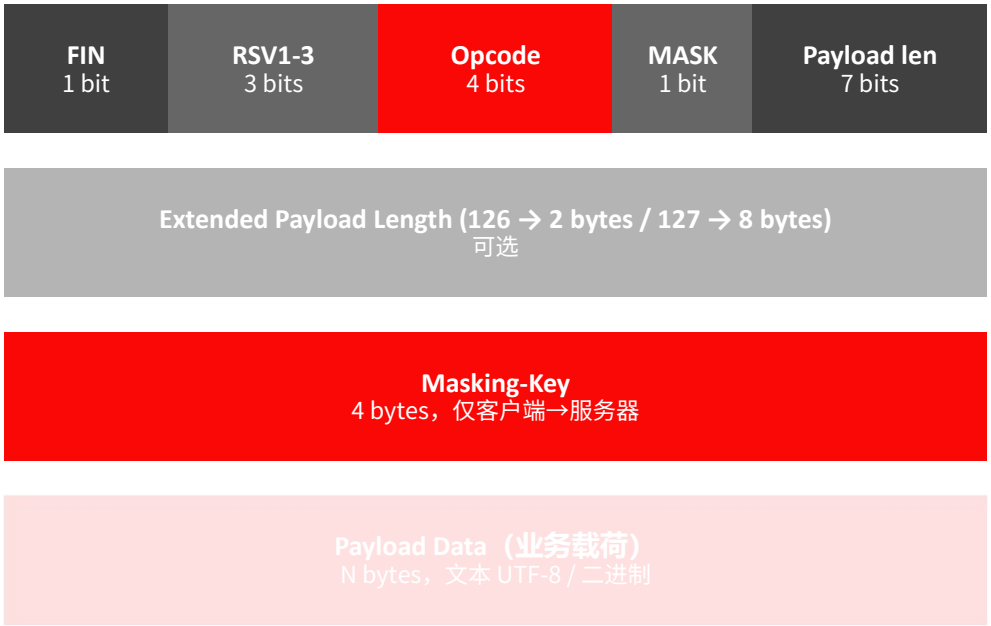
Part Two

从 TCP 建连升级到双向数据传输



数据帧格式

— 帧结构（RFC 6455）—



对比 HTTP：HTTP 头部 ≥ 500 字节，WebSocket 帧头仅 2-10 字节，高频小消息场景节省 95% 以上开销。

01 FIN 与 Opcode

FIN=1 表示这是消息最后一个片段；Opcode 0x1=文本、0x2=二进制、0x8=关闭、0x9=Ping、0xA=Pong。

02 Payload Length

7 位长度：0-125 直接表示；126 表示后续 2 字节为长度；127 表示后续 8 字节为长度，支持大消息。

03 Masking Key

RFC 要求客户端发出的帧必须带 4 字节随机掩码，载荷按 (i % 4) 异或解码，防止代理缓存投毒。

04 Payload Data

业务载荷：文本帧必须是合法 UTF-8；二进制帧可承载任意字节流，如 protobuf、图片、音频。

HTTP 与 WebSocket对比

01 连接方式
HTTP：短连接，每次请求建立、响应后断开
WS：持久连接，一次握手长期保持

02 通信方向
HTTP：客户端主动，服务器只能被动响应
WS：全双工，双方随时互推数据

03 协议头开销
HTTP：每次 500-800 字节头部
WS：帧头仅 2-10 字节

04 服务器推送
HTTP：不支持，需轮询/长轮询模拟
WS：原生支持，服务器主动下发

05 数据格式
HTTP：文本为主，每次带完整头部
WS：文本 / 二进制均可，帧头极简

06 适用场景
HTTP：网页浏览、REST API、表单提交
WS：聊天、实时行情、协同编辑、游戏

核心结论：HTTP 是"请求-响应"的文档协议，WebSocket 是"双向流式"的消息通道；两者不是替代关系，而是在同一端口上各司其职。

工程化实现

01 心跳与保活

Nginx、Netty、运营商 NAT 通常 60-120s 无消息就断开；应用层应每 20-30s 发 Ping/Pong 帧或业务心跳，保持连接活跃。

02 断线重连

网络抖动后自动重连，使用指数退避（1s→2s→4s…封顶 30s）加随机抖动，避免客户端同时重连冲击服务器。

03 鉴权与安全

握手时通过 Cookie、Authorization 头或 URL 上的 token 完成鉴权；生产环境必须用 wss://（TLS）；服务器校验 Origin 防跨站。

04 消息可靠性

WebSocket 本身不保证业务消息送达；关键消息需业务层 seq + ack + 重发；大文件/二进制分片走 HTTP，WS 只传指令。

一句话总结：WS 只解决"通道"问题，可靠性、鉴权、保活都要业务自己兜底——这是和 HTTP 最大的工程差异。

03 应用场景

Part Three

应用场景



01 即时通讯（IM）

消息秒级送达、已读回执、正在输入状态都需要服务器主动推送；弱网下快速重连。

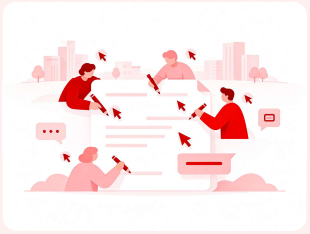
代表：*微信网页版、企业微信、客服系统、Slack*



02 实时行情与推送

股价、汇率、体育比分、新闻流需要毫秒级刷新；轮询在行情剧烈时延迟和带宽都不可接受。

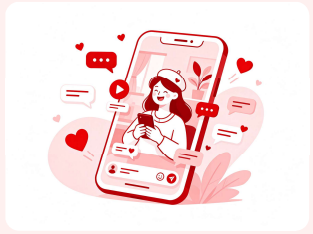
代表：*同花顺、东方财富、CoinDesk、推送通知*



03 协同编辑与白板

多人同时编辑文档，光标位置、选区、增量操作实时广播给所有协作者。

代表：*Google Docs、飞书文档、Figma、在线白板*



04 互动直播与游戏

弹幕、礼物、连麦、对战状态同步要求低延迟；二进制帧高效承载 protobuf 指令。

代表：*直播弹幕、联机对战、互动答题、语音房*

判断准则：需要服务器主动推送、消息频率高、对延迟敏感——就选 WebSocket；只是一次性取数据，用 HTTP 即可。

Thank you

获取更多产品信息, 请访问: www.quectel.com.cn
全国热线: 400 960 7678

移远全球中心
上海市松江区泗泾镇外婆泾路8号
联系电话：021 5108 6236　销售支持：sales@quectel.com
技术支持：support@quectel.com　电子邮件：info@quectel.com

QUECTEL

移远微信公众号