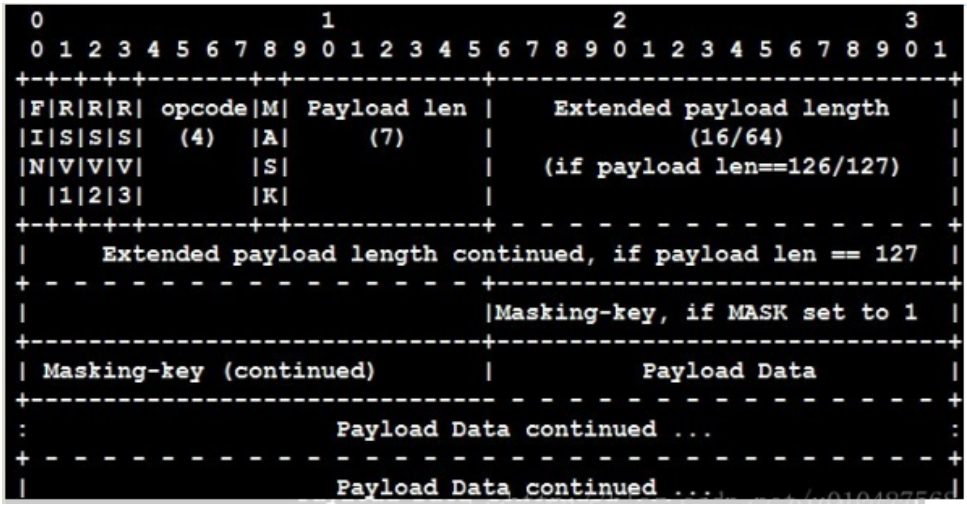
websocket协议整理

报文格式解析

1 图示



2 格式解析

2.1 FIN (1bit)

描述消息是否结束，为1说明结束，为0说明还有后续数据包

2.2 RSV1/RSV2/RSV3 (各1bit)

用于拓展定义，不拓展定义时必须为0

2.3 opcode (4bit)

- 0x0表示附加数据帧
- 0x1表示文本数据帧
- 0x2表示二进制数据帧
- 0x3-0x7时无定义，为以后的非控制帧保留
- 0x8表示连接关闭，接收到该帧即表示可以关闭底层的TCP连接了
- 0x9表示ping，当接收到该操作码的控制帧以后，应当立即发送一个包含pong操作码的响应帧，除非接收到了关闭帧。两端都会在连接建立后、关闭前的任意时间内发送Ping帧。Ping帧可以包含“应用数据”，Ping帧也可以作为keepalive心跳包。
- 0xA表示pong，当接收到该控制帧以后，知道对方还可响应。Pong帧必须包含与被响应Ping帧的应用程序完全相同的数据。如果终端接收到Ping帧，但还没对之前的Ping帧发送Pong响应，终端可以选择发送一个最近的Pong帧给最近处理的Ping帧。一个Pong帧可以被主动发送，作为单向心跳，但尽量不要主动发送Pong。
- 0xB-0xF暂时无定义，为以后的控制帧保留

2.4 MASK (1bit)

是否使用mask掩码处理。1表示使用，需要解码处理

2.5 payload length (7bit)

- 如果payload_length值是0-125，则是payload的真实长度
- 如果payload_length值是126，则后面2个字节形成的16位无符号整型数的值是payload的真实长度
- 如果payload_length值是127，则后面8个字节形成的64位无符号整型数的值是payload的真实长度

2.6 extend payload length(0/16/64 bit)

根据payload length的值确定所占位数，表示payload的长度

2.7 masking-key (32bit)

mask键值，用于对数据加密/解码

协议流程解析

1 websocket握手

在tcp三次握手之后，通过http/https进行websocket握手，将协议升级替换为websocket

4	0.006773	172.16.57.212	172.19.0.2	HTTP	293 GET /runtime/v1/recognize?res=aitmp&productId=12345_casr HTTP/1.1
5	0.006784	172.19.0.2	172.16.57.212	TCP	54 28002 → 49167 [ACK] Seq=1 Ack=240 Win=30336 Len=0
6	0.006990	172.19.0.2	172.16.57.212	HTTP	248 HTTP/1.1 101 Switching Protocols

1.1 升级协议使用的header

- Sec-WebSocket-Version：客户端发送，表示它想使用的WebSocket协议版本("13"表示RFC 6455)。如果服务器不支持这个版本，必须回应自己支持的协议版本
- Sec-WebSocket-Key：客户端发送，自动生成的一个键，作为对服务器的“挑战”，已验证服务器支持请求的协议版本
- Sec-WebSocket-Accept：服务器响应，包含Sec-WebSocket-Key的签名值，证明它支持请求的协议版本
- Sec-WebSocket-Protocol：用于协商应用子协议：客户端发送支持的协议列表，服务器必须只回应一个协议名，否则握手断开
- Sec-WebSocket-Extensions：用于协商本次连接要使用的WebSocket扩展：客户端发送支持的扩展，服务器通过返回相同的首部确认自己支持一个或多个扩展

1.2 客户端→服务端，握手请求

```
> Frame 4: 293 bytes on wire (2344 bits), 293 bytes captured (2344 bits)
> Ethernet II, Src: 02:42:dd:26:7e:f8 (02:42:dd:26:7e:f8), Dst: 02:42:ac:13:00:02 (02:42:ac:13:00:02)
> Internet Protocol Version 4, Src: 172.16.57.212, Dst: 172.19.0.2
> Transmission Control Protocol, Src Port: 49167, Dst Port: 28002, Seq: 1, Ack: 1, Len: 239
v Hypertext Transfer Protocol
  > GET /runtime/v1/recognize?res=aitmp&productId=12345_casr HTTP/1.1\r\n
    Upgrade: websocket\r\n
    Host: 10.12.8.13:38002\r\n
    Origin: http://10.12.8.13:38002\r\n
    Sec-WebSocket-Key: KgpykRcE0GQxFa7QEQeXAA==\r\n
    Sec-WebSocket-Version: 13\r\n
    Connection: Upgrade\r\n
    \r\n
    [Full request URI: http://10.12.8.13:38002/runtime/v1/recognize?res=aitmp&productId=12345_casr]
    [HTTP request 1/1]
    [Response in frame: 6]
```

1.3 服务端→客户端，握手响应。响应完成后就不再使用http连接，正式使用websocket连接

```
v Hypertext Transfer Protocol
  > HTTP/1.1 101 Switching Protocols\r\n
    Server: openresty/1.11.2.5\r\n
    Date: Wed, 27 Oct 2021 07:59:47 GMT\r\n
    Connection: upgrade\r\n
    Upgrade: websocket\r\n
    Sec-WebSocket-Accept: pWJ7cPIdJVEEQWyJfBuK8rKucmU=\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.000217000 seconds]
    [Request in frame: 4]
    [Request URI: http://10.12.8.13:38002/runtime/v1/recognize?res=aitmp&productId=12345_casr]
```

2 分帧规则

- 一个没有分片的消息由单个带有FIN位设置和一个非0操作码的帧组成
- 一个分片的消息由单个带有FIN位为0和一个非0操作码的帧，后面跟随零个或多个带有FIN为0和操作码设置为0的帧，且终止于一个带有FIN位设置且操作码为0的帧组成。

参考资料

1. https://blog.csdn.net/sermonlizhi/article/details/118609757
2. https://www.cnblogs.com/songwenjie/p/8575579.html