## [HTTP/2连接建立](http://www.blogjava.net/yongboy/archive/2015/03/18/423570.html)

HTTP/2协议在TCP连接之初进行协商通信，只有协商成功，才会涉及到后续的请求-响应等具体的业务型数据交换。

### HTTP版本标识符

* h2，基于TLS之上构建的HTTP/2，作为ALPN的标识符，两个字节表示，0x68, 0x32，即https
* h2c，直接在TCP之上构建的HTTP/2，缺乏安全保证，即http
* 在HTTP/2 RFC文档出现之前，以上版本字段需要添加上草案版本号，类似于h2-11,h2c-17

#### HTTP/2 请求过程

针对直接建立在标准TCP之上HTTP2，在未知服务器是否提供HTTP/2支持之前，可以依赖现有HTTP/1.1进行试探。

#### HTTP版本的请求内容

1. 客户端发起请求，只有请求报头：

GET / HTTP/1. 1Host: server. example. comConnection: Upgrade, HTTP2-SettingsUpgrade: h2c

HTTP2-Settings: <base64url encoding **of** HTTP/2 SETTINGS payload>

1. 服务器若不支持HTTP/2，直接按照HTTP/1.1响应即可

HTTP/1. 1 200 OK

Content-Length: 243

Content-Type: text/html

. . .

服务器支持HTTP/2，通知客户端一起切换到HTTP/2协议下吧

HTTP/1. 1 101 Switching Protocols**Connection**: Upgrade**Upgrade**: h2c

[ HTTP/2 connection . . .

1. 101响应空行之后，服务器必须发送的第一个帧为SETTINGS帧（其负载可能为空）作为连接序言
2. 客户端接收到101响应后，也必须发送一个序言作为响应，其逻辑结构：

PRI \* HTTP/2.0\r\n\r\nSM\r\n\r\n // 纯字符串表示，翻译成字节数为24个字节

SETTINGS帧 // 其负载可能为空

1. 服务器端和客户端所发送的连接序言有所不同。
2. 客户端可以马上发送请求帧或其它帧过去，不用等待来自服务器端的SETTINGS帧
3. 任一端接收到SETTINGS帧之后，都需要返回一个包含确认标志位SETTIGN作为确认
4. 其它帧的正常传输

#### HTTP/2的直接连接

客户端预先知道服务器提供HTTP/2支持，可以免去101协议切换的流程开销。 具体流程：

1. 客户端必须首先发送一个连接序言，其逻辑结构：

PRI \* HTTP/2.0\r\n\r\nSM\r\n\r\n // 纯字符串表示，翻译成字节数为24个字节

SETTINGS帧 // 其负载可能为空

1. 发送完毕序言之后，客户端可以不用等待来自服务器端响应，马上发送HTTP/2其它帧
2. 服务器端接收到客户端的连接序言之后，需要发送一个SETTINGS帧作为连接序言
3. 任一端接收到SETTINGS帧之后，都需要返回一个包含确认标志位SETTIGN作为确认
4. 其它帧的正常传输

#### HTTPS版本建立连接

HTTP/2安全版本在TLS上构建，协商采用的ALPN扩展协议，采用“h2”作为协议标识符（http版本则是“h2c”）。一定程度上可认为不存在试探是否支持或直接连接的烦恼，因为这个过程直接在TLS层协商而成。

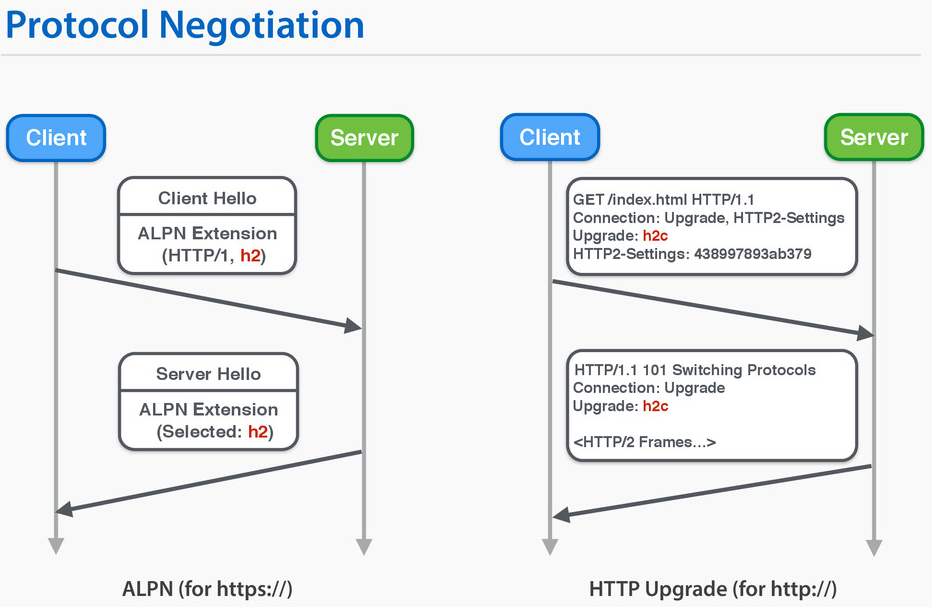
流程如下：

1. 客户端和服务器端TLS层协商
2. 客户端发送连接序言（同上表示，PRI + SETTINGS）
3. 接收到客户端连接序言之后，服务器端发送连接序言
4. 双方各自确认SETTINGS帧
5. 其它帧的正常传输

#### HTTPS和HTTP Upgrade方式协商

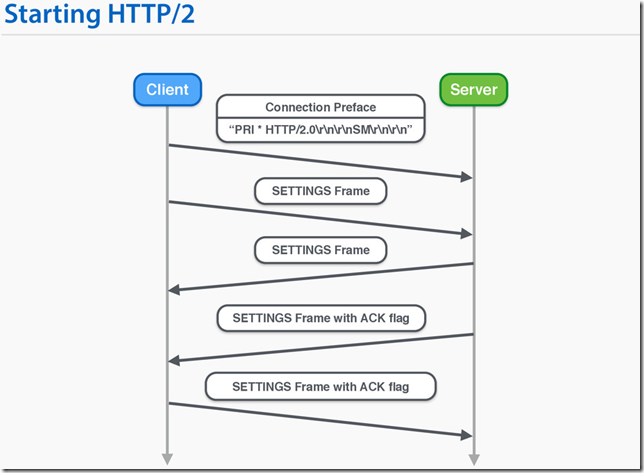
HTTPS协商是强制，封装在TLS之上ALPN扩展实现，HTTP只有非直接连接方式才会存在通过101 协议切换方式进行升级。

这里有一张图形象说明其流程。



#### 统一的连接过程

这里不论是HTTP还是HTTPS，在两端成功协商之后（或HTTP的直接连接），其连接过程都是一样的



### 注意事项

1. 客户端发起的HTTP/1.1请求，其流标识符为1，默认优先级；半关闭“half closed”状态，一旦完成HTTP/2的连接，将被应用于响应
2. 文档提到的客户端可以通过HTTP Alternative Services（简称为[ALT-SVC]，类似于CNAME机制）获得通知服务器是否支持HTTP/2，但目前看来仅仅是草案建议而已
3. 连接序言用于最后两端协商确认双方要使用HTTP/2协议，建立初始化的HTTP/2连接环境
4. 客户端若知服务器支持HTTP/2，可免去通过HTTP/1.1 101协议切换方式进行升级，在建立连接后即可发送序言，否则只能在接收到服务器端101响应后发送序言
5. 建立在TLS上的HTTP/2通过ALPN扩展协商机制取代101协议切换
6. 连接序言所包含的SETTINGS帧其负载可以为空
7. 针对一个TCP连接，服务器第一个要发送的帧必须是SETTINGS帧
8. 为了避免不必要延迟，客户端可以在发送完毕序言之后发送帧数据，不用等待来自服务器端的序言SETTINGS帧
9. 客户端接收到服务器端作为序言的SETTINGS帧，需要遵守其设定
10. 在一些环境下需要提供一个顺序机制，允许服务器在客户端发送业务帧之前发送SETTINGS，这需要客户端配合
11. 客户端和服务器端任何一方接收到无效连接序言需要抛出PROTOCOL\_ERROR类型连接错误，若收到GOAWAY帧，可忽略