**RPC协议底层原理与实现**

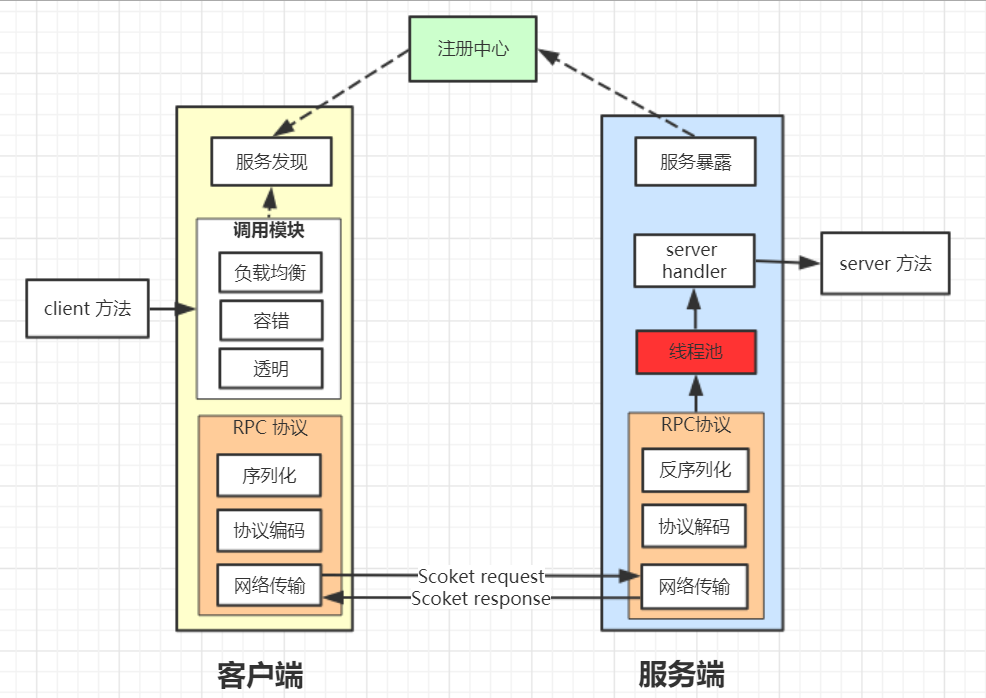
**课程概要：**

1. RPC协议基本组成
2. RPC协议报文编码与实现详解
3. Dubbo中所支持RPC协议使用

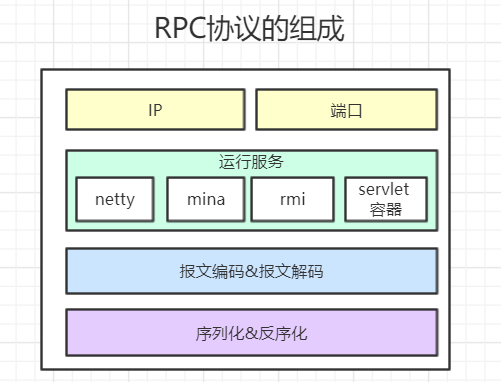
### **RPC协议基本组成**

### **RPC 协议名词解释**

在一个典型RPC的使用场景中，包含了服务发现、负载、容错、网络传输、序列化等组件，其中RPC协议就指明了程序如何进行网络传输和序列化 。也就是说一个RPC协议的实现就等于一个非透明的RPC调用，如何做到的的呢？

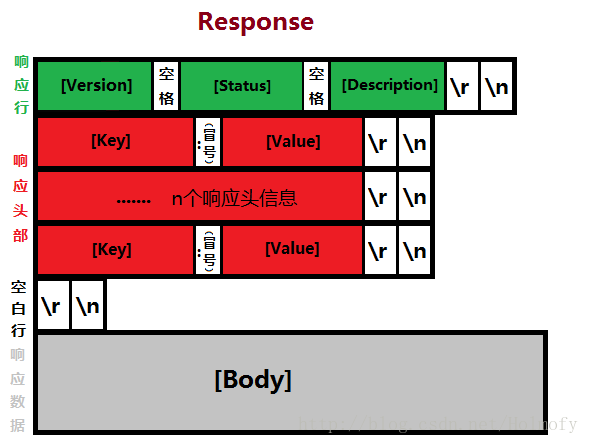


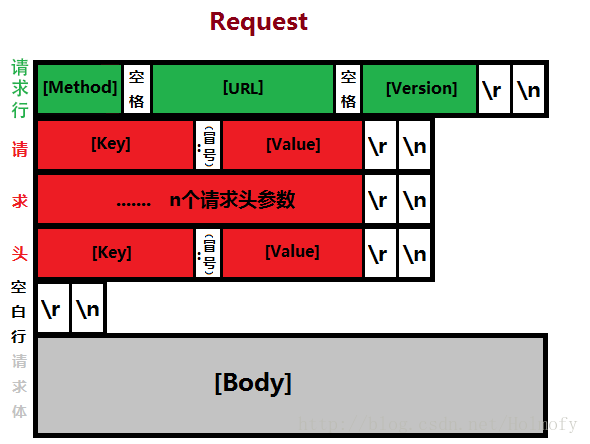
**协议基本组成：**



1. 地址：服务提供者地址
2. 端口：协议指定开放的端口
3. 运行服务:
4. netty
5. mina
6. RMI 服务
7. servlet 容器（jetty、Tomcat、Jboss）
8. 报文编码：协议报文编码 。注①:http 报文编码 。注②:Dubbo 报文编码
9. 序列化方式：
10. Hessian2Serialization、
11. DubboSerialization、
12. JavaSerialization
13. JsonSerialization

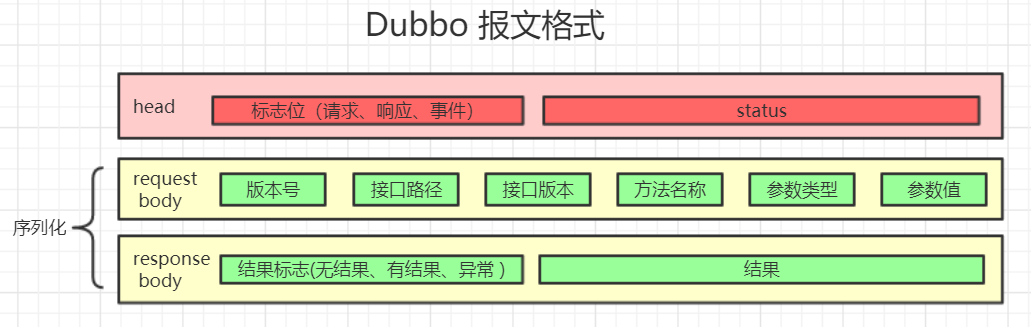
### **RPC协议报文编码与实现详解**





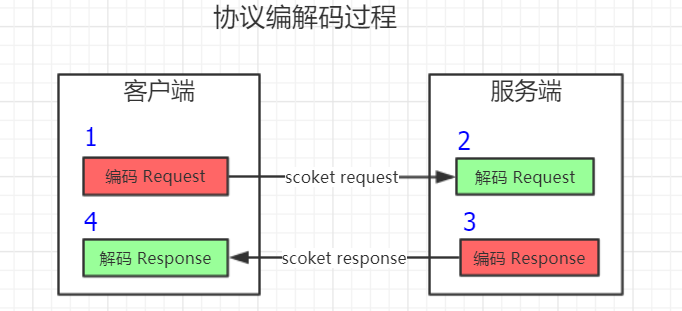
(注①：http 报文编码)

**注②Dubbo 协议报文编码：**



*（注：相关源码参见 com.alibaba.dubbo.rpc.protocol.dubbo.DubboCodec）*

**协议的编解码过程：**



**Dubbo 协议编解码实现过程** *(源码来源于dubbo2.5.8 )*

1、DubboCodec.encodeRequestData() 116L // 编码request

2、DecodeableRpcInvocation.decode() 89L // 解码request

3、DubboCodec.encodeResponseData() 184L // 编码response

4、DecodeableRpcResult.decode() 73L // 解码response

### **Dubbo中所支持RPC协议使用**

**dubbo 支持的RPC协议列表**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
| 名称 | 实现描述 | 连接描述 | 适用场景 |
| dubbo | 传输服务: mina, netty(默认), grizzy 序列化: dubbo, hessian2(默认), java, fastjson 自定义报文 | 单个长连接 NIO异步传输 | 1、常规RPC调用 2、传输数据量小 3、提供者少于消费者 |
| rmi | 传输：java rmi 服务 序列化：java原生二进制序列化 | 多个短连接 BIO同步传输 | 1、常规RPC调用 2、与原RMI客户端集成 3、可传少量文件 4、不支持防火墙穿透 |
| hessian | 传输服务：servlet容器 序列化：hessian二进制序列化 | 基于Http 协议传输， 依懒servlet容器配置 | 1、提供者多于消费者 2、可传大字段和文件 3、跨语言调用 |
| http | 传输服务：servlet容器 序列化：http表单 | 依懒servlet容器配置 | 1、数据包大小混合 |
| thrift | 与thrift RPC 实现集成，并在其基础上修改了报文头 | 长连接、NIO异步传输 |  |

***关于RMI不支持防火墙穿透的补充说明：***

原因在于RMI 底层实现中会有两个端口，一个是固定的用于服务发现的注册端口，另外会生成一个***随机***端口用于网络传输。这个随机端口就不能在防火墙中提前设置开放开。所以存在*防火墙穿透问题*

### **协议的使用与配置:**

Dubbo框架为了更灵活扩展，其支持多种协议，用户只需要在 provider 应用中 配置即可*<*dubbo:protocol*>* 元素即可。

<!--

name: 协议名称 dubbo|rmi|hessian|http|

host:本机IP可不填，则系统自动获取

port：端口、填-1表示系统自动选择

server：运行服务 mina|netty|grizzy|servlet|jetty

serialization：序列化方式 dubbo|hessian2|java|compactedjava|fastjson

[详细配置参见dubbo 官网 dubbo.io](http://dubbo.io/books/dubbo-user-book/references/xml/dubbo-protocol.html)

-->

<dubbo:protocol name="dubbo" host="192.168.0.11" port="20880" server="netty"

serialization=“hessian2” charset=“UTF-8” />

#TODO 演示采用其它协议来配置Dubbo

dubbo 协议采用 json 进行序列化 (源码参见：com.alibaba.dubbo.rpc.protocol.dubbo.DubboProtocol*)*

采用RMI协议 (源码参见：*com.alibaba.dubbo.rpc.protocol.rmi.RmiProtocol)*

采用Http协议 (源码参见：*com.alibaba.dubbo.rpc.protocol.http.HttpProtocol.InternalHandler)*

采用Heason协议 (源码参见:com.alibaba.dubbo.rpc.protocol.hessian.HessianProtocol.HessianHandler)

[pom.xml](https://dn-shimo-attachment.qbox.me/TaswKTTNBpI8mJmy/pom.xml)