# Network Resource Lookup Protocol

Version 1.0

Drafted by Shulai Zhang

Contributors: Qiaoqiao Li, Bingjue Sun.

## Specific Words Explanation

Resource:

In this article, it refers to the data that is provided by some services and requested by other in a certain network. There are static resources(e.g., file, constant, etc.), as well as dynamic resources(e.g., HTTP session object, outcome of a called function, etc.).

Resource Provider:

In this article, it refers to the system that provides some resource in a certain network. It could be a service or application written in whatever language.

Resource Requester:

In this article, it refers to the system that requests some resource in a certain network.

Be aware that a system could be both a resource provider and a resource requester at the same time.

Central Message Service:

In this article, it refers to the service responsible for transfer message between systems in a certain network. It is the Central Message Service that makes all the systems in the network can communicate with each other.

## Protocol Description

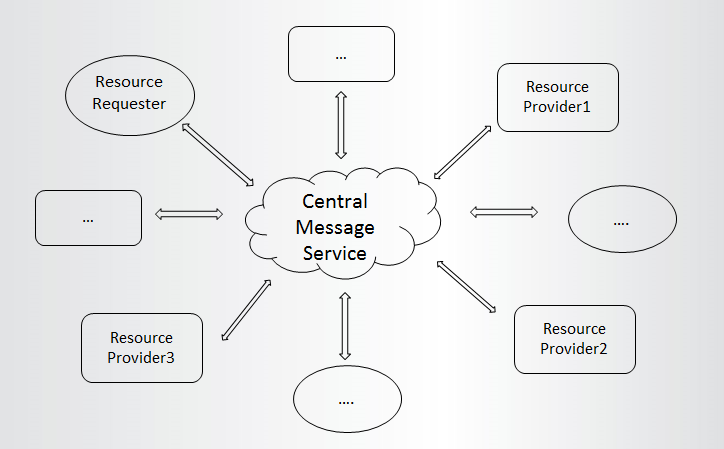


Figure1

This protocol is based on a existing message service, so that is a priority. It is usually implemented to handle tasks in multi-application environment, as well as clustered environment.

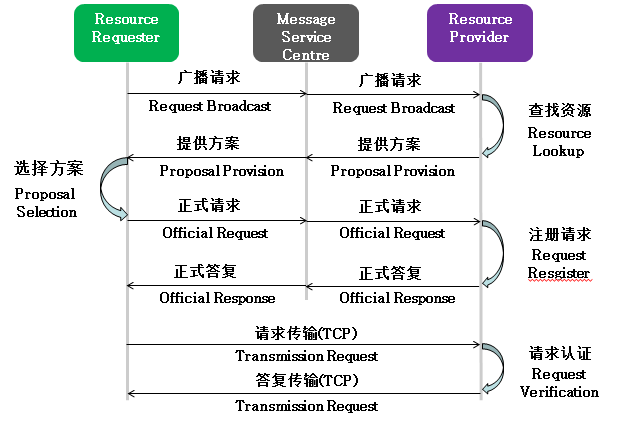


Figure 2. the Workflow View

Messages

## Protocol Details

#### Message Format

Every message is composed by two parts, message head and variables.

#### Message Head

All NRLP message should begin with a 8-byte message head.

Including:

* Protocol Name. UTF-8 encoded. 6 bytes total.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Description | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |
| Byte 1 | Length MSB (0) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Byte 2 | Length LSB(4) | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Byte 3 | ‘N’ | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| Byte 4 | ‘S’ | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| Byte 5 | ‘L’ | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| Byte 6 | ‘P’ | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |

* Protocol Version. 1 byte.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|  | Protocol Version | | | | | | |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

* Message Type. 1 byte.

|  |  |
| --- | --- |
| Byte Value | Message Type |
| 0 | Reserved |
| 1 | Request Broadcast |
| 2 | Proposal Provision |
| 3 | Official Request |
| 4  5 | Official Response  Transmission Request |
| 6 | Final Response |

#### Variables

Client ID:

Every single instance of application in this network should have a unique ID, so it makes others know who is actually speaking. It should be UTF-8 encoded in the message body.

Request Code:

It is sensible logic that a requester can send multiple request at the same time. So it is the requester's responsibility to keep them identical. The simple way is give each one of them a unique code. This code should be reusable when a request has done. It is a 2 bytes unsigned integer, so the maximum size is 65535.

Transmission Code:

Correspond to the request code, there is a transmission code in the provider side. It is generated once a transmission is confirmed. That is after official response. The implementation is the same as request code, 2 bytes unsigned integer.

Resource Name:

Each resource should have a unique name in this network, so it can be picked up when remote client searches with name. UTF-8 encoded.

Resource Type:

Every resource is marked to a single type, so when client gets the resource it can handle it separately regarding to its type. It is a 2 bytes unsigned integer.

|  |  |
| --- | --- |
| value | Resource type |
| 0 | Static Resource. When requester broadcasts a request with this code, the provider should look up the resource from all resources that have resource type between 0 to 29,999. |
| 1 | File Resource. |
| 2～9999 | Reserved. |
| 10000～29999 | Customization available. |
| 30000 | Dynamic Resource. When requester broadcasts a request with this code, the provider should look up the resource from all resources that have resource type between 30,000 to 65535. |
| 30001 | HTTP Session |
| 30002 | Java Serializable |
| 30003～39999 | Reserved. |
| 40000～65535 | Customization available. |

Transmission Type:

It is the way both requester and provider have agreed how to transmit the data. 1 byte long, Up to 255 transmission possibilities.

|  |  |
| --- | --- |
| value | Transmission type |
| 0 | One-shot transmission. |
| 1 | Resume-able transmission |
| 2 | Unique-maintained transmission |
| 3～99 | Reserved. |
| 100～255 | Custom usage. |

Resource Size:

Resource size should be included in the Proposal Provision. This part could be a quite large number. It is encoded using Multi-Byte Integer algorithm.

(Not Complete yet! )

#### Message Details

广播请求(Request Broadcast) ： RR向局域网内广播资源请求。（在mqtt作为MQ Protocol的情况下， 该消息会以qos 0来发送。默认主题为 nrlp/res/req/broadcast。

消息体:

1. 协议名。 UTF-8编码。共6 比特。

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Description | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |
| Byte 1 | Length MSB (0) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Byte 2 | Length LSB(4) | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Byte 3 | ‘N’ | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| Byte 4 | ‘S’ | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| Byte 5 | ‘L’ | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| Byte 6 | ‘P’ | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |

1. 协议版本。1 比特。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|  | Protocol Version | | | | | | |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

3. 消息类型。1 比特。

|  |  |
| --- | --- |
| 代码 | 消息类型 |
| 0 | 预留 |
| 1 | 广播请求 |
| 2 | 提供方案 |
| 3 | 请求资源 |
| 4  5 | 请求答复  开始请求 |
| 6 | 传输开始 |

4. 资源请求者在该网络内的唯一标示。暂定和mqtt协议的client id保持一致。长度为1-23位char。UTF-8编码。

5. 请求号。2 byte。

6. 请求资源名称。UTF-8编码。暂定为1-65535个char。在该网络内，资源名称必须唯一。

7. 资源类型。 两个比特。0-65535种可能的资源类型。其中，0-29999均为静态资源(static resource)， 30000-65535均为动态资源(dynamic resource)。

|  |  |
| --- | --- |
| 代码 | 类型名称 |
| 0 | 静态资源。当广播请求消息中使用该代码，资源提供方应在所有类型为0-29999的资源中进行搜寻。 |
| 1 | 文件资源。 |
| 2～9999 | 预留代码。 |
| 10000～29999 | 项目自定义静态资源类型。 |
| 30000 | 动态资源。当广播请求消息中使用该代码，资源提供方应在所有类型为30000～65535的资源中进行搜寻。 |
| 30001～39999 | 预留代码。 |
| 40000～65535 | 项目自定义动态资源类型。 |

7. 可选参数标识位。一个比特位。是否含有请求传输方式和方案监听渠道。如果有，需要标识位置一。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  | Way of Transmission Flag | Channel of Proposal listening Flag |
|  | x | x | x | x | x | x | 0 | 0 |

7. 请求传输方式。一个比特位。提供了高达255种传输要求的可能性。

|  |  |
| --- | --- |
| 代码 | 传输类型 |
| 0 | 直接传输 |
| 1 | 断点续传 |
| 2 | 唯一传输 （传输完的同时，本地会删除，保证全网络资源的唯一性） |
| 3～99 | 预留代码 |
| 100～255 | 自定义传输要求 |

请求审核(Request Verification) : RP在接收到RR的请求广播后，首先会检查自己是否拥有请求中描述的资源，然后再审核这个请求者是否有请求该资源的权限，如果有则准备提供方案。

提供方案(Proposal Provision)

1. 协议名。

2. 协议版本。

3. 消息类型。

4. 请求号

5. 资源名称

6. 资源类型。两个比特。

6. 资源大小。资源大小通过以下算法encode。其中X表示要编码的十进制数字。

编码逻辑：

do

digit = X MOD 128

X = X DIV 128

// if there are more digits to encode, set the top bit of this digit

if ( X > 0 )

digit = digit OR 0x80

endif

'output' digit

while ( X> 0 )

其中MOD的是余数算法(在c语言里为 : %)，DIV 为整数除法 (c语言里为 /)，OR 为字节或算法(c语言里的 |)。

解码逻辑：

multiplier = 1

value = 0

do

digit = 'next digit from stream'

value += (digit AND 127) \* multiplier

multiplier \*= 128

while ((digit AND 128) != 0)

其中 AND 是字节且算法(c语言里的 &)

当运算结束， value即为该要表示的数字。

7. 资源提供者的客户id。1-23位char。utf-8编码。在该网络中唯一。

8. 提供的传输方式。

9. 提供的传输时速。2 比特。单位为： kb/s。

选择方案(Proposal Selection) ： 当 资源请求者 在等待一定时间的 提供方案 后，从得到的方案中，选择最适合自己的方案。这种选择条件，根据需求不同，可能受下面因素影响，如：方案提供者身份，方案提供的传输方式，传输时速等。

请求资源(Resource Request) : 当 资源请求者 在确定请求方案后，向 资源提供者 发送 请求资源的指令。

1. 协议名。

2. 协议版本。

3. 消息类型。

7. 资源请求者的客户ID。

7. 请求号。2 bytes

4. 资源名称。用于做资源比对。

5. 资源类型。用于做资源比对。

6. 资源请求的传输方式。

请求注册(Request Registration) ： 当 资源提供者 接收到 资源请求者 的请求后，必要时，再一次根据请求者的客户ID和请求的资源名称和类型进行一次验证，如果验证通过，则进行一个请求注册动作，即将该客户ID和资源ID 一起注册为一个传输任务。如果验证不通过， 则返回信息拒绝请求。

请求答复(Request Permission) : 当 资源提供者 就 资源请求者 的请求，进行必要的审核和准备后，正式给 资源请求者 请求路径 和 传输任务的ID，允许 资源请求者 开始资源的传输，或者 审核未通过，拒绝客户的传输请求。

1. 协议名。

2. 协议版本。

3. 消息类型。

4. 请求号。 2bytes

6. 资源提供者的客户id。

7. 资源传输任务的ID。由 资源提供者 在其自身的context内生成，在其自身内保持唯一。每个资源传输任务的ID对应一个资源传输任务，资源传输任务包括 资源请求者，资源名称和资源类型，以及传输方案和传输时速。UTF-8编码。

8. 资源提供TCP的地址。UTF-8。

开始请求(Transmission Request ) : 当 资源请求者 拿到请求答复，从中获取解析出，资源传输任务的ID和资源提供的TCP地址。资源请求者 发起TCP请求。

1. 协议名。

2. 协议版本。

3. 消息类型。

1. 资源传输任务的ID。
2. 剩余长度。标识了请求参数的长度。用multi-byte integer。
3. 请求参数。

传输资源(Resource Transmission) : 当 资源提供者 接收到 开始请求 后，通过资源传输任务的ID，找到需要进行传输的资源，按之前协议好的速率进行传输。

1. 协议名。

2. 协议版本。

3. 消息类型。

1. 请求号
2. 资源长度
3. 资源