Network Resource Lookup Protocol

网络资源查询获取协议

Version 1.0

Drafted By Shulai Zhang

名词定义：

网络资源(Resource)：在特定网络中，某个或多个服务提供的资源，可以为静态资源（例如，文件），可以是动态资源（例如一个HttpSession）。

Resource: In this article, it means 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.).

资源提供者(Resource Provider): 在特定网络中，提供网络资源(Resource) 的服务对象。可以是一个Web Service，也可以是一个进程程序，只要其有资源提供给这个网络。

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

资源请求者(Resource Requester): 在特定网络中，寻求网络资源(Resource) 的对象。以下简称RR。

Resource Requester: In this article, it means the system that requests for some resource in a certain network.

注：一个对象可能即为资源提供者(Resource Provider) 同时又是资源请求者(Resource Requester)。

A system could be both a resource provider and a resource provider at the same time.

消息服务中心(Message Service Broker)：在特定网络中，资源请求者(Resource Requester) 和资源提供者(Resource Provider)之间沟通的桥梁。（本例以mqtt broker为例。）

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

协议描述：

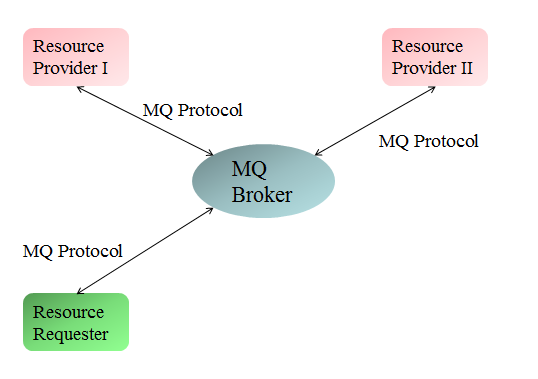


Figure 1. 网络描述

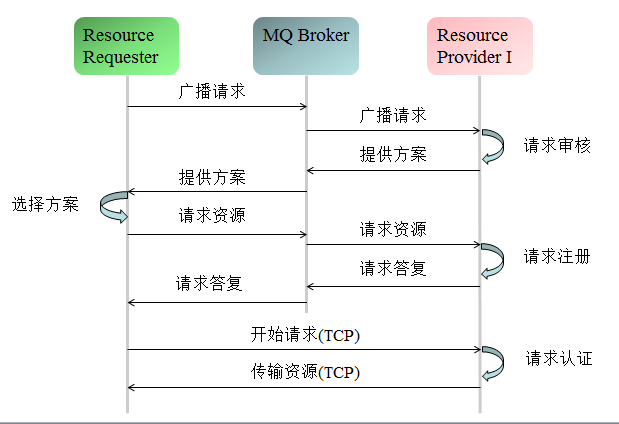


Figure 3. NRLP流程

协议详解

Protocol Detail

消息结构

Message Format

每个消息都有两部分组成。消息头 和 参数。

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

消息头

Message Head

所有NRLP消息都必须带有消息头。共八个比特。

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

其中包括：

协议名。UTF-8编码。共六个比特。

Protocol name: UTF-8 encoded. 6 bytes.

协议版本。一个比特。

Protocol version. 1 byte.

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

消息类型。 一个比特。

Message type. 1 byte.

|  |  |
| --- | --- |
| 代码 byte value | 消息类型 message type |
| 0 | 预留 reserved |
| 1 | 广播请求 Request Broadcast |
| 2 | 提供方案 Proposal Provision |
| 3 | 正式请求 Official Request |
| 4  5 | 正式答复 Official Response  请求传输 Transmission Request |
| 6 | 答复传输 Final Response |

广播请求： 资源请求者向局域网内广播资源请求。

Request Broadcast: The resource requester broadcasts his demand over the network.

参数：

1. 资源请求者在该网络内的唯一标示。UTF-8编码。

2. 请求资源名称。UTF-8编码。在该网络内，资源名称必须唯一。

3. 资源类型。两个比特。最多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。 |

4. 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 |

5. 方案监听渠道。UTF-8编码。

variables:

1. Requester Id: Unique Id of this requester in the network. UTF-8 encoded.

2. Resource Name: Unique Name of the requested resource in the network. UTF-8 encoded. If there are multiple resources with same name in the network, then for sure they are the copies of the named resource.

3. Resource Type: 2 bytes. Up to 65,535 resource type possibilities. The bytes value between 0 to 29,999 denotes a static resource, and 30,000 or more denotes a dynamic resource.

|  |  |
| --- | --- |
| 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 | Custom usage for static resource types. |
| 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. |
| 30001～39999 | Reserved. |
| 40000～65535 | Custom usage for dynamic resource types. |

4. Transmission Type: 1 byte. Up to 255 transmission type possibilities.

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

5. Proposal-listening Channel. UTF-8 encoded.

查找资源 : 资源提供者在接收到资源请求者的请求广播后，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。

Resource Look-up: After receiving the broadcast from requester, the provider will look the resource up in its repository, if it exists, then an optional authority check is applicable. Finally, a proposal of providing this resource is given.

提供方案

Proposal Provision

参数：

1. 资源名称。用于做资源比对。

2. 资源类型。用于做资源比对。

3. 资源大小。用Multi-Byte Integer来编码。

4. 资源提供者的唯一标识。utf-8编码。在该网络中唯一。

5. 提供的传输方式。

6. 提供的传输时速。2 比特。无符号整数。单位为： kb/s。

选择方案

NRLP and UTF-8

UTF-8是一种高效的string编码格式。在NRLP中，string会在其编码前增加两个比特位，记录该string的比特长。

UTF-8 is an efficient encoding of Unicode character-strings that optimizes the encoding

of ASCII characters in support of text-based communications. In NRLP, strings are prefixed with two bytes to denote the length, as shown in the table below.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **bit** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| byte 1 | String Length MSB | | | | | | | |
| byte 2 | String Length LSB | | | | | | | |
| byte 3 ... | Encoding Character Data | | | | | | | |

注意，这里使用的是string的比特长，而不是string的character数。

下面是协议名(NRLP)的编码例子。

Be aware that here we use the byte length of the string, not the number of characters.

For example, NRLP is encoded in UTF-8 as shown in table below.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **bit** | **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 |