

Hot Protocol Version 1

Hottentot RPC Framework

Kamran Amini

May 9, 2016

Contents

1	Introduction	2
2	Glossary	2
3	Request	2
3.1	Request Type	3
4	Future Features	3

1 Introduction

This document talks about request and response structures and mechanisms in Hottentot RPC Framework. Purpose of this protocol is to convey Method Invocation request and response. Current protocol is serialization transparent and can convey a method call with arguments produced with different serialization algorithms. In this version, Hottentot's runtimes can only work with internal serialization mechanism.

2 Glossary

SERIALIZATION

A process in which an object turns into a byte array to be transferred in a channel.

STRUCT

A term used for encapsulation of fields related to a specific entity. It is a structure and will be generated for each programming language differently.

IDL (INTERFACE DEFINITION LANGUAGE)

An IDL is a language transparent to all programming languages which Hottentot supports. IDL can be generated to any target languages supported by Hottentot RPC Framework.

HOT FILE

A file which contains IDL. Hot files usually end with `.hot` extension.

GENERATOR

A tool for generating stub and struct source codes for a target programming language. Currently, generators for C++ and Java languages are available.

RUNTIME

A library for a specific programming language which performs Service and Proxy operations. Currently, runtimes are only available for C++ and Java.

ENDPOINT

Endpoint is a combination of IP address and a port. A service object is bound to an endpoint.

SERVICE

Service is an object serving method invocation requests.

PROXY

Proxy is an object which produces method invocation requests and receives the response. A proxy object directly talks to a service object.

PROXY SIDE

.

SERVICE SIDE

.

3 Request

A Method Invocation request consists of following fields:

- Request Type (1 Byte)
- Service Id (4 Bytes)

- Method Id (4 Bytes)
- Number of Arguments (1 Byte)
- Arguments as an array of LV Structures. (Variable Length)

Figure 1: Structure of a Method Invocation request.

Figure1: Hot Protocol V1 Request Structure

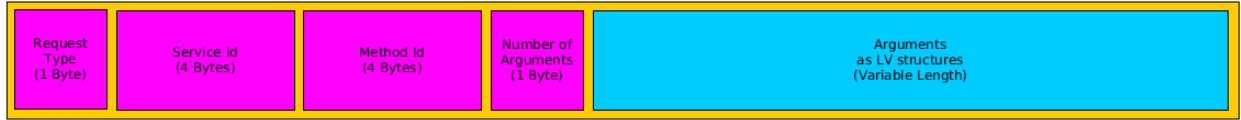
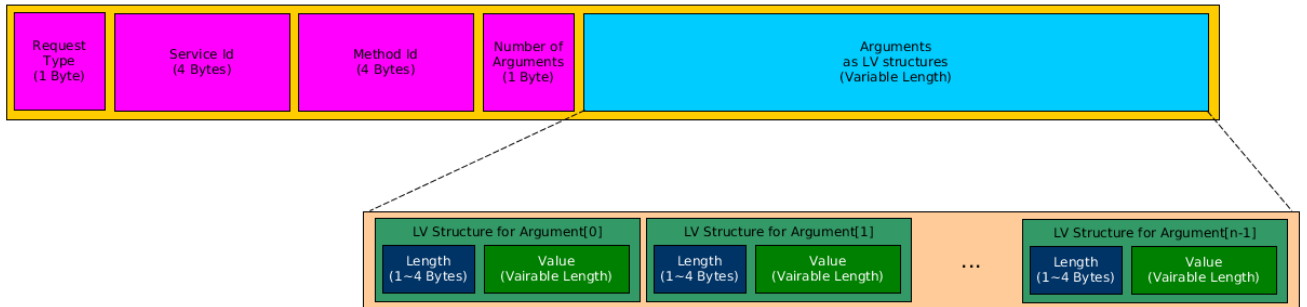


Figure 1 shows the structure of a request. Arguments can be transferred using LV structures. LV structures include Length and Value. Maximum length for a single LV structure is $2^{32} - 1$ since length values can be stored in a field at most 4 bytes. Figure 2 shows the LV structures in detail.

Figure 2: Structure of a Method Invocation request with arguments' LV structures.

Figure2: Hot Protocol V1 Request Structure and LV Structures



Value can be any byte array but usually it is a serialized object. Serialization method can be anything. Hottentot itself provides an algorithm for serialization and `serialize()` and `deserialize()` methods are generated for every struct. Current stub generation mechanism works only with Hottentot's internal serialization.

3.1 Request Type

Determines the request type. Following C enum shows the probable values for this field.

```

enum RequestType {
    Unknown = 0,
    ServiceListQuery = 1,
    InvokeStateless = 2,
    InvokeStateful = 3
};
  
```

Different values can be:

- **Unknown:** It means nothing to Hottentot service side and these requests should be ignored by the implementation.
- **ServiceListQuery:** Proxy side queries about the list of available services at service side.

4 Future Features

- Version should be added to request and response structures.