SDP协议分享

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 文档名称 |  | 版本号 | 1.00.00 | | |
| 文档编号 |  | | |
| 文档类别 | 使用说明 | 文档阶段 | 初稿 | | |
| 项目名称 | TWS | 作者 | 何继胜 | | |
| 承担部门 | 软件研发部 | 批准 |  | | |
| 文档日期 | 2019/7/10 | 使用范围 | 公司内部 | 页数 | 3 |

目录

[1 文档版本 3](#_Toc13644387)

# 文档版本

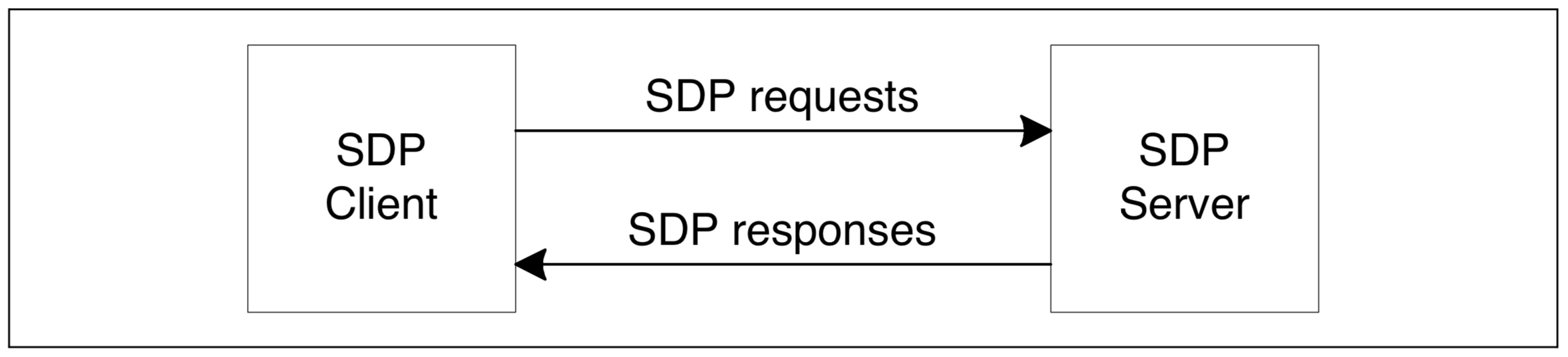
|  |  |  |
| --- | --- | --- |
| 日期 | 版本号 | 简要描述 |
| 2019/7/4 | 1.00.00 | 初稿 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# 服务概述

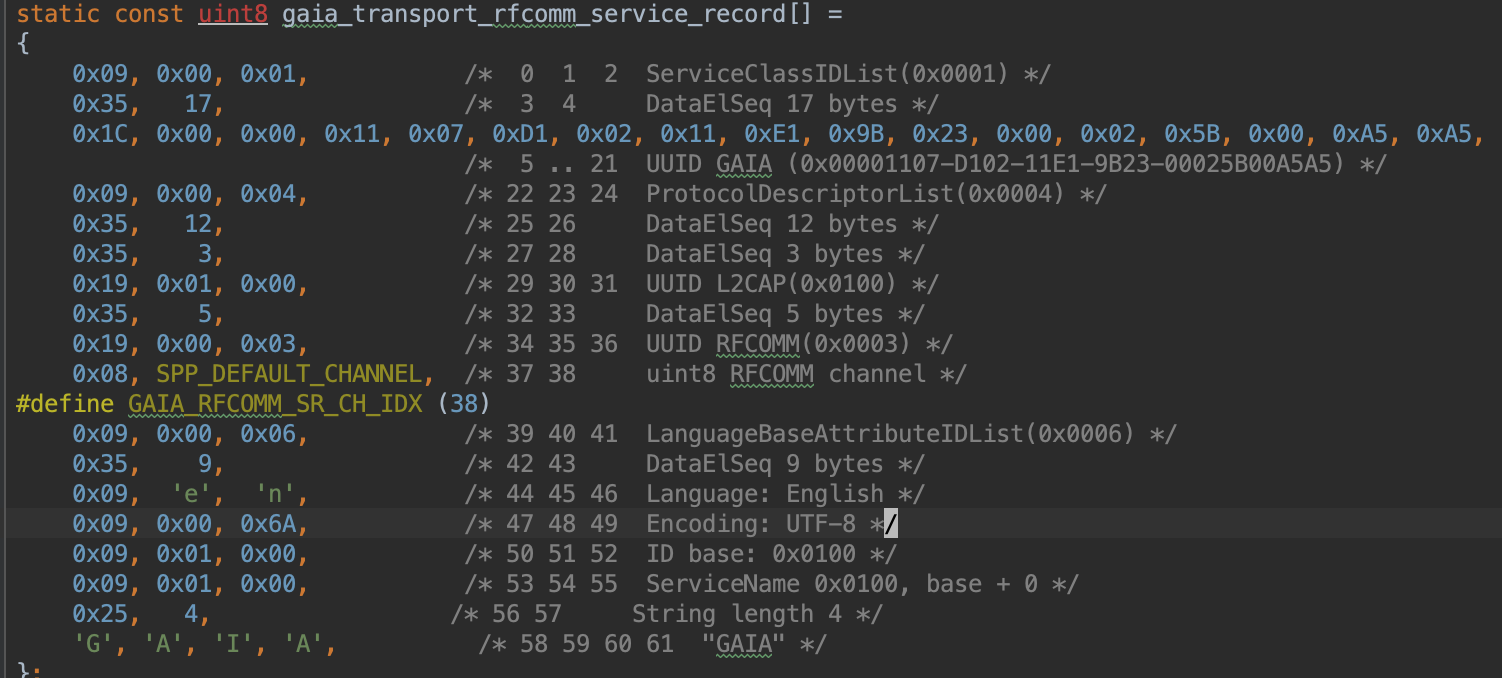
SDP, Service Discovery Protocol，服务发现协议。

## 概念

SDP提供了一种用于发现服务及这些可用服务属性的方法，但它不提供利用这些服务的机制。其架构是Client-Server模式，如下图所示：



SDP Server维护了一个服务记录(Service Record)列表，每个条目包含了该服务的信息。 如：

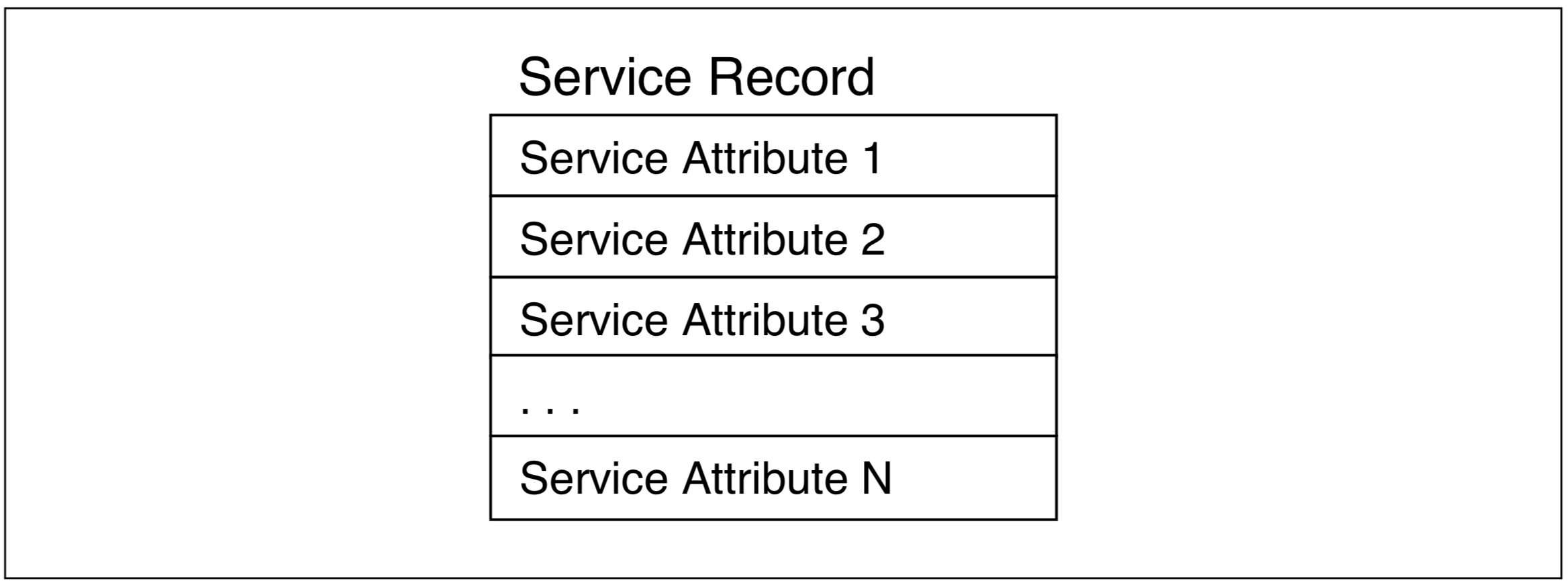


SDP Client通过SDP request来向SDP Server获取服务记录信息。Client可以通过打开一条单独的连接来使用Server提供的某种服务。

当Server的服务改变时，Client必须通过其他方式来了解这一信息，以便能够通过SDP来查询 。同时，当Server由于某种原因不可用时，Client可以使用SDP轮询Server 。当Server不再响应请求时，Client可以推断该Server不可用。

## 服务记录(Service Record)

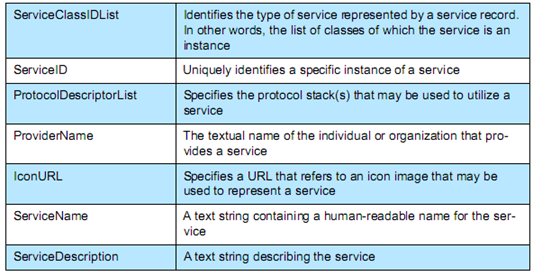
每一个Service用Service Record来表示。 每一个Service Record由若干Service Attribute组成，如下图所示：

****

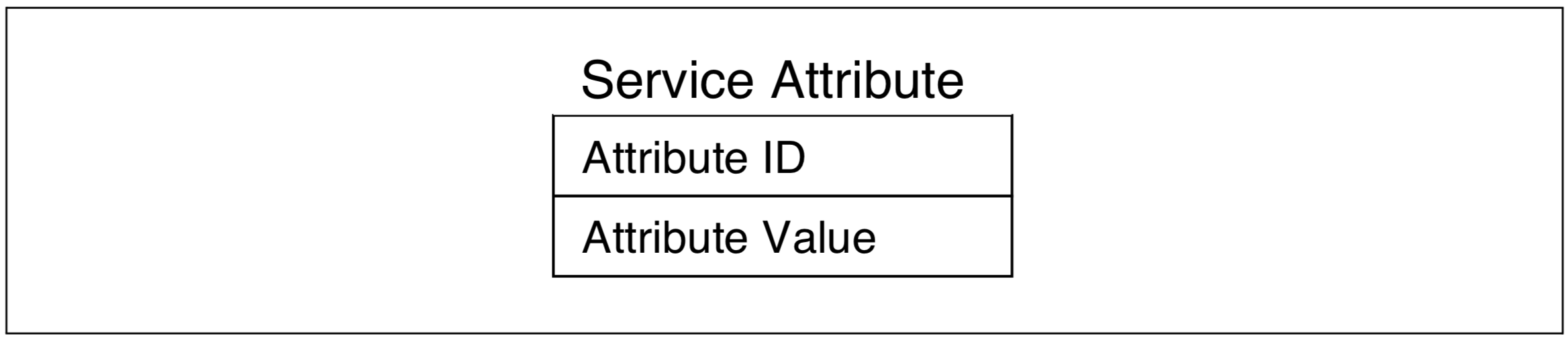
在SDP Server中，用Service Record Handle来唯一标识一个Service Record ，而Service Record Handle则使用32-bit数字表示。在Service Record List中，Server使用0x00000000来表示SDP本身。

## 服务属性(Service Attribute)

每个服务属性描述了一个服务的单个特征，实例如下：

****

一个服务属性包含了两个部分: 属性ID和属性值：



属性ID是16-bit无符号整型，用以区分Server中不同属性，属性ID还确定了相关的属性值的语义，属性值字段长度是可变的，由关联属性ID和服务记录类别决定。

## 服务类(Service Class)

每个服务是一个服务类的实例 ，服务类定义了包含在该类服务记录的所有属性。每个属性定义了指定的属性ID，及使用的属性值和属性值得格式

服务记录包含了特定服务类及通用服务类的属性。每个服务记录都包含一个唯一的ServiceClassIDList属性，该属性描述了本服务的服务类型。

每个服务类被分配了唯一的标识符 ，这个服务类标识符包含在ServiceClassIDList属性的属性值中，被称为UUID。

下面来看一个实例，下面这个SDP recored来源于ADK的sink例程中的SPP服务：

在上面的SDP Service Record中，ServiceClassIDList**属性ID**对应的**属性值**为：SerialPort (0x1101)，这就是SPP服务对应的SDP标识符。客户端通过查找该标识符，来获取服务端是否支持SPP服务。其中ServiceClassIDList，ProtocolDescriptorList，L2CAP，RFCOMM，ServiceName是通用服务类属性，SerialPort Version是SPP服务特定的服务属性(私有属性)。

某个服务的特定(私有)属性如何查找？。

通常属性ID和属性值封装在不同的数据单元(Data Element)里面，因此属性ID和属性值之间需要插入DataElSeq数据单元，用来记录属性值所占据字节数，这样服务器在解析SDP数据记录时，能够有效定位该属性记录的边界。

上述的SDP记录用jeason表示为：

|  |
| --- |
| ServiceClassIDList(0x0001){  SerialPort(0x1101);  }  ProtocolDescriptorList(0x0004){  L2CAP(0x0100){  RFCOMM(0x0003){  SPP\_DEFAULT\_CHANNEL;  }  }  }  LanguageBaseAttributeIDList(0x0006) {  0x656e,  0x006a,  0x0100  }  BluetoothProfileDescriptorList(0x0009){  SerialPort(0x1101);  }  SerialPortVersion {  }  ServiceName(0x0100){  SERVICENAME("SPP Dev");  } |

## 服务查找

服务查找允许Client基于包含服务记录的属性值，来获取特定服务记录的服务记录句柄(Service Record Handle)。 当一个SDP Client有某个服务记录句柄时，它可以请求特定的属性值。SDP不提供基于任意属性值的服务记录查找，只提供基于UUID的查找 。

## 服务浏览

SDP提供了基于服务类共享属性机制来浏览服务，这个属性被称为BrowseGroupList，Client通过创建一个包含代表根浏览组的UUID的服务搜索模式来浏览Server的服务。

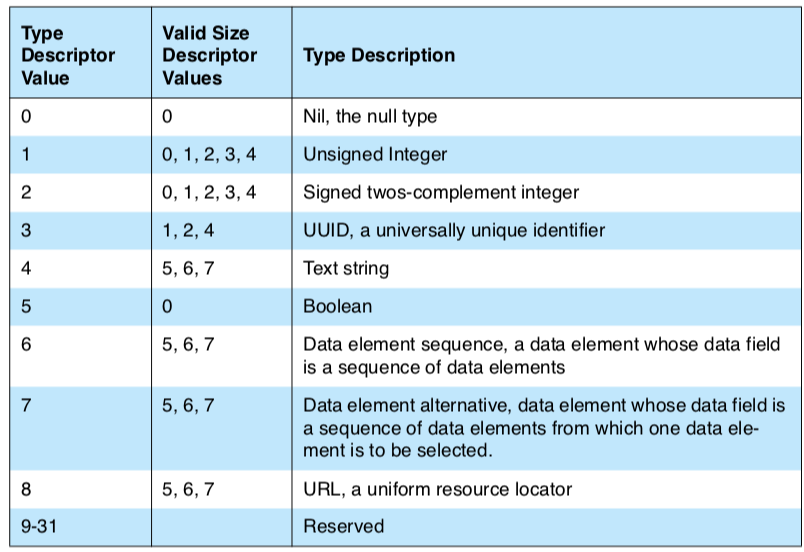
## 数据表示(物理存储形式)

SDP的使用数据单元(DataElement)来表示数据(属性ID，属性ID范围，属性值)。数据单元是一种类型化的数据表示，它由两个字段组成:首部字段(Header Field)和数据字段(Data Field)。

首部字段包含两个部分: 类型描述符(Type Descriptor)和大小描述符(Size Descriptor) 。数据字段是一个字节序列，其长度由大小描述符指定，其含义则由类型描述符指定。

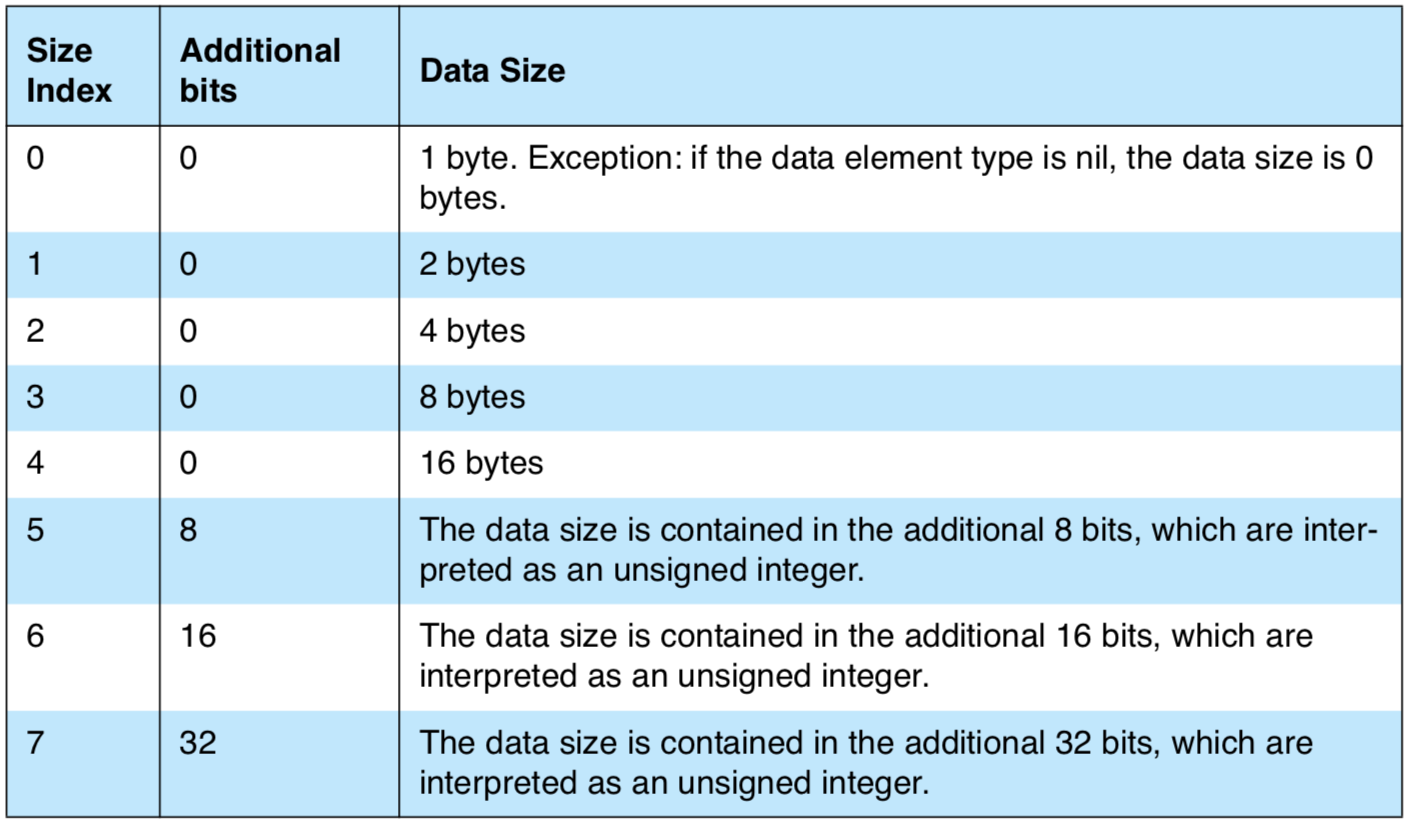
### 类型描述符

数据单元的类型使用5-bit的类型描述符来表示，它包含在首部字段第一个字节的高五位。下面是已经定义的类型：

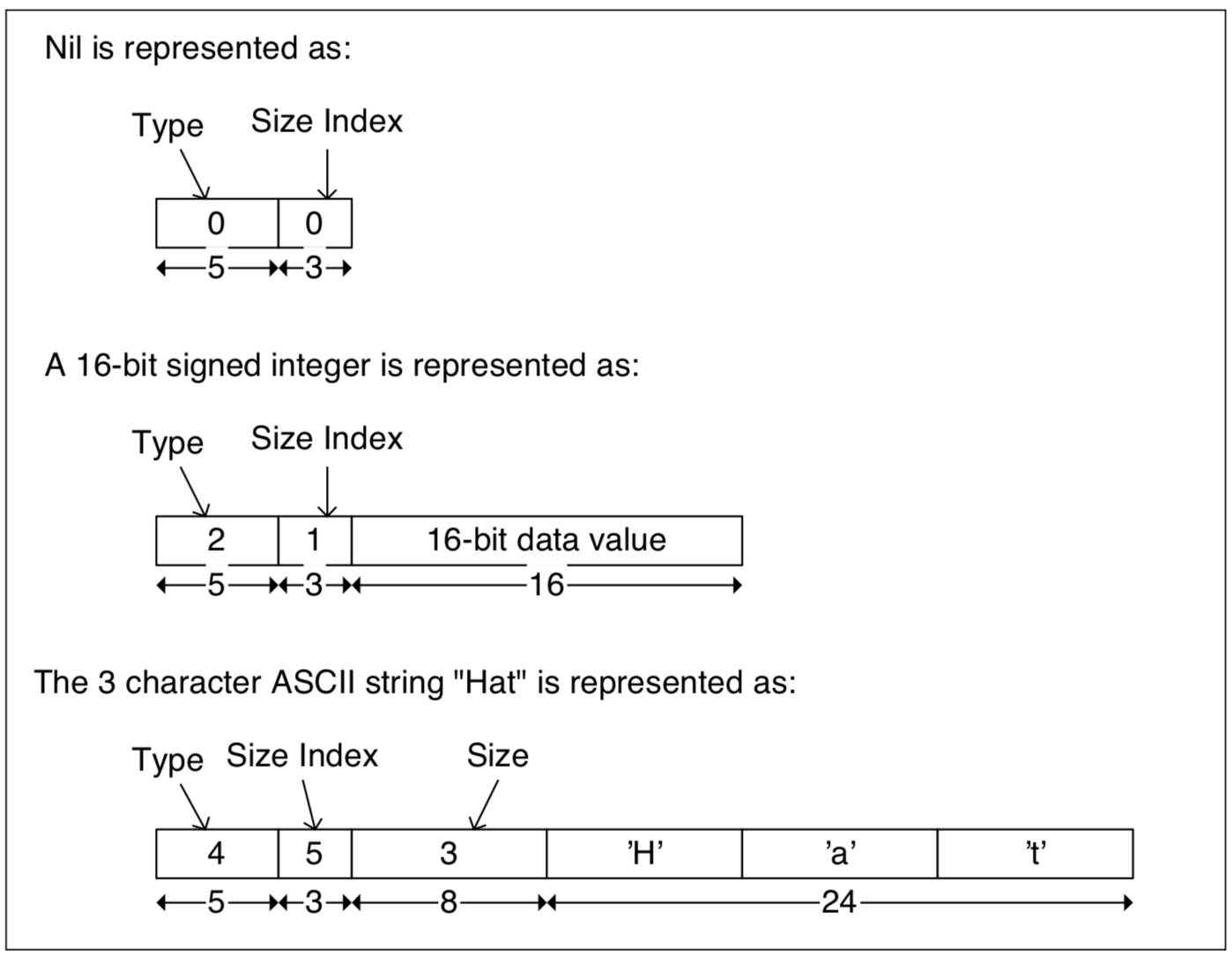


### 大小描述符

数据单元的大小描述符包含在首部字段第一个字节的低三位 ，它表示为的大小指数，其后为0/8/16/32bits ，大小指数的编码如下：

****

### 数据单元实例

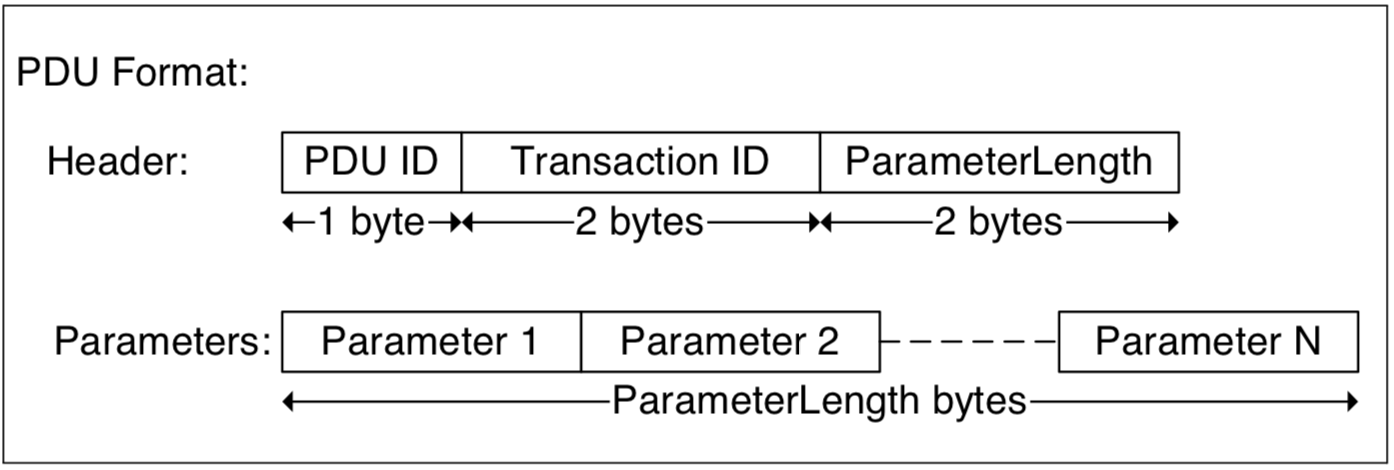


# 协议说明

SDP使用Request/Response模型 ，其中每个事务(Transaction)包含一个请求协议数据单元(PDU)和一个响应PDU，SDP使用L2CAP作为传输协议，在建立连接并发出SDP Request后在给定的时间内，只有收到该Request的Response后，才能发出其他的Request，传输采用Big-Endian，高位先低位后的方式。

## PDU格式

SDP PDU包含一个Header和Parameters，Header包含三个字段: PDU ID, Transaction ID, Parameter Length。

各参数解释：

PDU ID: Size: 1 Byte

|  |  |
| --- | --- |
| **Value** | **Parameter Description** |
| N | The PDU ID field identifies the type of PDU. I.e. its meaning and the specific parameters. |
| 0x00 | Reserved |
| 0x01 | SDP\_ErrorResponse |
| 0x02 | SDP\_ServiceSearchRequest |
| 0x03 | SDP\_ServiceSearchResponse |
| 0x04 | SDP\_ServiceAttributeRequest |
| 0x05 | SDP\_ServiceAttributeResponse |
| 0x06 | SDP\_ServiceSearchAttributeRequest |
| 0x07 | SDP\_ServiceSearchAttributeResponse |
| 0x07-0xFF | Reserved |

TransactionID: Size: 2 Bytes

|  |  |
| --- | --- |
| **Value** | **Parameter Description** |
| N | The TransactionID field uniquely identifies request PDUs and is used to match response PDUs to request PDUs. The SDP client can choose any value for a request’s TransactionID provided that it is different from all outstanding requests. The TransactionID value in response PDUs is required to be the same as the request that is being responded to. Range: 0x0000 – 0xFFFF |

ParameterLength: Size: 2 Bytes

|  |  |
| --- | --- |
| **Value** | **Parameter Description** |
| N | The ParameterLength field specifies the length (in bytes) of all parame- ters contained in the PDU. Range: 0x0000 – 0xFFFF |

# Sdpcontrol

