

基于 TTCN-3 的 CoAP 一致性测试

杜泓庆, 黄小红, 何嘉杰

(北京邮电大学 网络技术研究院 信息网络中心, 北京 100876)

摘 要: 协议一致性测试是确保协议实现与协议标准相一致的重要保障, 有助于协议实现的进一步完善。CoAP (constrained application protocol) 即受限应用协议是特别为受限节点和受限网络设计的通用应用层协议。CoAP 目前仍处于草案阶段, 对 CoAP 进行了分析, 并基于标准化测试语言 TTCN-3 设计实现了 CoAP 一致性测试系统, 并使用该系统对 CoAP 实现进行了测试。

关键词: CoAP; 一致性测试; TTCN-3; 编解码器; 适配器

中图分类号: TP393.06

文献标识码: A

文章编号: 1000-436X(2013)Z2-0090-04

CoAP conformance testing based on TTCN-3

DU Hong-qing, HUANG Xiao-hong, He Jia-jie

(Network Information Center, Institute of Networking Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: Protocol conformance test is of great importance in making sure that the protocol implementations conform to the protocol specification, thus helps protocol implementer to improve the implementations. CoAP (constrained application protocol) is a generic Web application specified for constrained nodes and constrained network which is currently an Internet draft. A CoAP conformance test suite based on TTCN-3, a standardized testing language, was designed, implemented and used after analyzing the protocol.

Key words: CoAP; conformance test; TTCN-3; codec; adapter

1 引言

在物联网中有很多节点, 出于能量、空间、成本等因素的考虑, 某些节点所能使用的资源受限, 如很多传感器节点只有很小的内存和很弱的处理能力; 同时如 6LoWPAN^[1]等受限网络只有有限的分组大小, 网络的吞吐量有限。受限节点和受限网络的存在, 限制了这些受限环境可能支持的网络协议的复杂性, 传统网络应用中的 HTTP 协议并不适用于这些环境。因此 IETF 成立了 CoRE (constrained RESTful environments) 工作组, 并制定了 CoAP, 目的在于实现受限环境中面向资源的应用框架。

CoAP 还处于草稿阶段, 目前最新的版本是

draft-ietf-core-coap-18^[2]。CoAP 的一致性测试不仅能够测试协议实现与协议标准是否一致, 而且对于正在制定中的协议也能提供可供改进的参考。自从 CoRE 工作组提出 CoAP 草案以来, CoAP 便受到学界与工业界的高度关注。伴随物联网进一步的发展与普及, 更多 CoAP 相关的研究将会开展, 更多 CoAP 相关的产品将会推出。

与所有协议一样, CoAP 是由自然语言定义的, 这往往导致网络设备在协议实现上出现差异。因此, 为提高设备产品质量, 减少故障率, 提高互联互通性, 除了进一步修订协议标准外, 还需要对各种协议实现进行严格的一致性测试, 以验证是否符合协议规范。本文研究填补了 CoAP 一致性相关研究的空白, 将推动 CoAP 的发展和应用。

收稿日期: 2013-09-06

基金项目: 国家 CNGI 专项“可演进的下一代高智能网络架构研究和试验”基金资助项目

Foundation Item: China Next Generation Internet (CNGI) Project: Research and Trial on Evolving Next Generation Network Intelligence Capability Enhancement(NICE)

2 相关背景

2.1 一致性测试与 TTCN-3

不同系统之间正常通信的实现，首先就是要求通信系统的行为满足协议规范的要求。协议实现必须保证它能够按照协议标准执行，即与标准一致。确保协议实现与协议标准一致的测试过程被称为协议一致性测试 (protocol conformance testing)。一致性测试属于功能测试也叫做黑盒测试，与被测系统内部结构无关。目前实际使用的一致性测试标准是基于国际标准组织 ISO 的 IS-9646 标准,即国际电信联盟 ITU 的 ITU-T X29x 标准^[3]。该标准定义了由自然语言描述的协议一致性测试的方法和框架。

TTCN 是由欧洲电信标准化协会 (ETSI) 制定和维护的。TTCN-3(testing and test control notation version 3)^[4,5]测试和测试控制表示第三版于 2000 年发布。TTCN-3 成功地用在了很多技术的认证上,如 IPv6、WiMax、3GPP 等。

2.2 CoAP

CoAP 的交互和 HTTP^[6]的客户端/服务器模式相似。和 HTTP 不同的是 CoAP 的交互是在数据报传输方式如 UDP 上异步传输的。CoAP 消息的基本交换对于请求/响应交互是透明的。CoAP 定义了 4 种消息: CON、NON、ACK、RST。CoAP 请求使用 GET、PUT、POST 和 DELETE 等方式,服务器响应代码也和 HTTP 响应代码类似。

为了降低解析的复杂性,CoAP 的消息用简单的二进制格式编码。消息头部定长度 4 byte,包含版本、消息类型、令牌长度、请求或响应代码、消息编号等信息。头部之后是可变长度的令牌,用于匹配请求和响应、个数可变的选项、可选的载荷。

3 基于 TTCN-3 的 CoAP 一致性测试系统

3.1 测试系统结构

基于 TTCN-3 的测试系统如图 1 所示。顶层是 TTCN-3 的测试脚本,底层是被测系统(system under test)。在 TTCN-3 测试脚本和被测系统之间包括了运行时层和平台层的代码,主要是编解码器模块和适配器模块,此外还有控制模块和日志模块等。

3.2 测试套设计

3.2.1 测试例生成

以 CoAP 为基础,分析提取出协议中的测试点,

形成 CoAP 一致性测试套。CoAP 中主要的通信实体是客户端和服务器,它们之间的通信模式为请求/响应模式,典型的过程为:客户端发出请求消息,服务器经过处理后返回相应的响应消息。所有的通信都是通过 CoAP 消息的传递完成的,消息格式也是 CoAP 一致性测试的重要环节。整个测试套的基本功能点,包括消息格式、消息交互、请求生成/理解、回复生成/理解等。

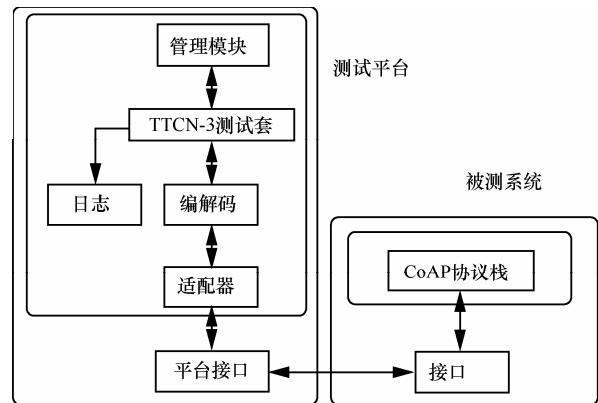


图 1 基于 TTCN-3 的测试系统

根据通信实体,可以将测试套分为服务器和客户端的测试,然后再根据消息格式、交互流程将服务器和客户端的测试分为更细的分组。为了验证协议实现的可靠性,还可以加入部分无效测试。

3.2.2 基于 TTCN-3 的测试套设计

TTCN-3 测试脚本主要包括测试数据定义和具体的测试逻辑,是测试平台的核心,测试套包含所有测试例,用 TTCN-3 语言描述。首先要根据 CoAP 的消息格式,定义出测试中使用到的 CoAP 消息结构,测试中使用模板(template)定义具体的消息,模板有两方面作用,其一是用于传送一个特定值的集合,其二是测试接收的值的集合是否与模板定义匹配。

消息结构使用 record 进行嵌套定义。如 CoAP 消息定义如下:

```

type record CoAPMessage {
    CoAPHeader header,
    CoAPToken token,
    CoAPOption option,
    Payload payload optional
}
  
```

该结构将 CoAP 消息按其格式定义为头部、令牌、选项、载荷的集合,具体的每一个域再进一步

根据其结构进行定义。如 CoAP 消息 CoAPMessage 结构中的头部结构,即 CoAPHeader 可以定义如下:

```
type record CoAPHeader {
    bitstring CoAPVersion length(2),
    bitstring CoAPType length(2),
    bitstring CoAPTokenLength length(4),
    octetstring CoAPCode length(1),
    octetstring CoAPMessageID length(2)
}
```

类似地, CoAP 头部结构也是一个集合,分别由版本、类型、令牌长度、代码、消息号等组成。这些结构由 TTCN-3 的基本类型如比特串或者字节串类型定义。

测试套定义了大量在测试中使用的消息模板,如发送消息头部对应的模板示例为

```
template CoAPHeader RequestHeader := {
    CoAPVersion := '01'B,
    CoAPType := '01'B,
    CoAPTokenLength := '0001'B,
    CoAPCode := '01'O,
    CoAPMessageID := '7d34'O
}
```

该模板中, CoAP 消息头部的版本、类型、令牌长度、代码、消息号都填入了具体的值,可用于参数化测试系统发送的消息或者对接收消息的匹配。

在只考虑 client 和 server 的情况下, CoAP 的交互流程和拓扑结构比较简单。因此测试例的逻辑结构也比较简单。一个典型的测试例的实现包括这样的流程: 测试例定义所需要的变量, 对端口进行操作, 发送数据分组, 开启定时器, 等待回复, 对接收到的数据分组进行解析和逻辑判断, 若接收数据为空或定时器超时则判定结果 fail; 若数据分组正确则关闭定时器, 并判断是否需要继续发送数据或者判定结果 pass。

3.3 编解码模块

编解码模块负责测试数据的编码和解码工作, 这些数据包括可执行测试套和被测系统之间通信的数据^[7]。

在测试工具开发过程中需要针对协议规范制定特定的测试数据。和 HTTP、SIP 协议等文本描述的协议不同, CoAP 使用了二进制的消息格式。根据 CoAP 消息格式的特点, 主要使用到的简单数据类型包括比特串(bitstring)、字节串(octetstring)、

字符串(charstring) 等, 消息结构使用了 record 进行嵌套定义。

编解码模块中使用到了 2 个接口函数: tciEncode() 负责编码, tciDecode() 负责解码。

tciEncode() 函数是编码模块的入口函数, 调用 encode() 函数进行具体编码。encode() 函数内部主要为 switch-case 结构, 首先判断编码类型, 然后调用具体的编码函数进行编码。具体的编码函数部分定义了 3 个编码函数 encode_bitstring()、encode_octetstring()、encode_charstring(), 分别对应了比特串、字节串、字符串 3 种类型。

tciDecode() 函数是解码模块的入口函数, 调用 decode() 函数进行具体的解码。decode() 根据 CoAP 消息分为头部、令牌、选项、载荷 4 部分, 依次调用对应的解码函数(decode_header()、decode_token()、decode_option())进行解码, 最后返回解码结果。与编码部分对应, 具体的解码是由 decode_bitstring()、decode_octetstring()、decode_charstring() 3 个函数来对比特串、字节串、字符串 3 种基本类型进行解码的。

3.4 适配器模块

适配器模块分为系统适配器和平台适配器。

系统适配器主要完成可执行测试套与被测系统间的网络通信功能, 用于发送和接收数据。由于 CoAP 通常采用的是无连接的 UDP, 系统适配器模块使用 UDP 通信, 通过套接字编程来实现, 使用 sendto 发送数据, 使用 recvfrom 等待接收数据。

平台适配器主要提供了计时器功能和一些无法由 TTCN-3 代码实现的外部函数等^[8]。

3.5 管理和日志模块

管理模块为用户提供友好一致的软件操作界面, 方便协议测试人员通过完成协议测试的开始、查看测试结果、查看运行日志等系列操作。管理模块设计着重考虑到的有 2 点: 一是易用性, 二是可移植性。基于上述考虑, 管理模块设计采用 HTML 搭建基本框架, 结合 PHP 脚本添加互操作性功能的浏览器/服务器模式。

日志模块在执行测试例时同步生成 HTML 格式的测试日志, 测试人员可以用浏览器直接查看, 也可以在 TTCN-3 测试平台中查看。

4 测试执行与分析

TTCN-3 语言编写的程序经过编译后生成 C 语

言程序，进一步和编解码模块、适配器模块一起编译生成可执行的测试文件。被测目标选取了 libcoap^[9] 这个 CoAP 的开源实现。libcoap 的发行版本是基于 draft-ietf-core-coap-03^[10] 实现的，和最新的 CoAP 有相同的地方，也有不同的地方，可以预期测试例可能部分通过。



图 2 Web 方式的测试控制界面

被测 libcoap 服务器运行在 Linux 主机，测试系统运行在 Windows 主机，之间通过 IP 网络相连。在实际的测试中，测试例有的能够通过，有的不能，基本符合预期。进一步查看日志、抓分组分析可以看到该一致性测试系统能够根据测试过程正确判定测试例结果，针对 CoAP 设计的测试套基本覆盖 CoAP 消息通信中的关键测试点，能够满足一致性测试的基本需求。

5 结束语

本文在研究 CoAP 草案基础上，对相关理论及行业应用进行了深入，设计并开发 CoAP 一致性测试系统。该测试系统将会为 CoAP 实现提供验证，以帮助开发者进一步完善 CoAP 应用，为 CoAP 的发展起到推动作用。

CoAP 作为未来物联网的其中一种重要应用层协议，也处于不断修改与快速完善的标准化进程之中。随着研究的深入，将会有更多新的特性被添加到 CoAP 中，也会有不少旧的特性在新版本的标准中被修改或删除。本课题需要在继续跟踪 CoAP 修订的基础上，继续完善下面几个部分：

- 1) 设计并实现更多的测试例，以覆盖 CoAP 新的特性；
- 2) 完善系统适配器模块，支持 6LoWPAN 通信；
- 3) 进一步完善系统界面，以允许测试人员进行部分测试用例的选取与执行。

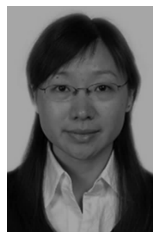
参考文献：

- [1] MONTENEGRO G, KUSHALNAGAR N, HUI J, *et al.* RFC 4944 - Transmission of IPv6 Packets over IEEE 802.15.4 Networks[S]. 2007.
- [2] SHELBY Z, HARTKE K, BORMAN C. Constrained Application Protocol (CoAP) Draft-Ietf-Core-Coap-18[S]. 2013.
- [3] ITU-T X 290, OSI Conformance Testing Methodology and Framework-General Concepts[S]. 1995.
- [4] ETSI ES 201 873-1 V4.4.1 Methods for Testing and Specification(MTS); The Testing and Test Control Notation version 3; Part 1: TTCN-3 Core Language[S]. 2012.
- [5] WILLCOCK C, DEIB T, TOBIES S, *et al.* An Introduction to TTCN-3 [M]. England: John Wiley & Sons, 2005.
- [6] FIELDING R, GETTYS J, MOGUL J, *et al.* RFC 2616 - Hypertext Transfer Protocol--HTTP/1.1[S]. 1999.
- [7] 宁姣. 基于 TTCN-3 的通用编码器的研究与实现[D]. 北京: 北京邮电大学, 2012.
NING J. Research and Implementation of Generic Codecs Based on TTCN-3[D]. Beijing: Beijing University of Posts and Telecommunications, 2010.
- [8] 葛岩. 支持多种协议的 TTCN-3 适配器的设计与实现[D]. 北京: 北京邮电大学, 2009.
GE Y. Design and Implementation of TTCN-3 Adapters for Multiple Protocols[D]. Beijing: Beijing University of Posts and Telecommunications, 2009.
- [9] Libcoap: C-implementation of CoAP[EB/OL]. <http://libcoap.sourceforge.net/>, 2013.
- [10] SHELBY Z, FRANK B, STUREK D. Constrained Application Protocol (CoAP) Draft-Ietf-Core-Coap-03[S]. 2010.

作者简介：



杜泓庆(1988-), 男, 四川资中人, 北京邮电大学硕士生, 主要研究方向为协议测试等。



黄小红(1978-), 女, 广东河源人, 博士, 北京邮电大学教授, 主要研究方向为下一代互联网关键技术。



何嘉杰(1987-), 男, 广东广州人, 硕士, 主要研究方向为协议测试等。