

IPv6 协议一致性测试描述语言设计与实现

张晓东,董平,苏伟

(北京交通大学电子信息工程学院,北京 100044)

摘 要:IPv6 协议一致性测试是检验 IPv6 协议实现是否正确的重要方法。为弥补我国在该领域研究水平的不足,文中提出了一种模块化测试描述模型,设计了模块化测试描述语言(Modular Test Description Language, MTDL),并开发了基于该模型和语言的 IPv6 协议一致性测试系统。在对 IPv6 协议标准深入分析的基础上设计了一套测试集,用自主开发的测试系统在 Linux 系统平台上对 IPv6 协议进行了测试,得到了预期结果,验证了 MTDL 语言的正确性和有效性。

关键词:IPv6;一致性测试;模块化测试描述语言;测试系统

中图分类号:TP31

文献标识码:A

文章编号:1673-629X(2013)07-0040-04

doi:10.3969/j.issn.1673-629X.2013.07.010

Design and Implementation of Description Language of Conformance Test for IPv6

ZHANG Xiao-dong, DONG Ping, SU Wei

(College of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China)

Abstract:IPv6 conformance test is an important method to check the correctness of an IPv6 implementation. To make up for the lack of the level of research in the field in China, proposed a modular test description model, designed a modular test description language (MTDL), and developed the IPv6 conformance test system based on the model and the language. Based on in-depth analysis of the IPv6 protocol specification, designed a test set and tested on the Linux platform using the self-developed test system. Got the expected results, which verified the correctness and effectiveness of the MTDL.

Key words:IPv6; conformance test; MTDL; test system

0 引言

一致性测试是协议测试领域的重要方面,是一种“黑盒测试^[1]”,即不涉及协议的内部实现,只从外部行为来检验协议实现与协议标准的符合程度,是协议测试的重要方法。在协议的一致性测试中,首先需要选择合适的测试方法,确定测试结构,其次是严格按照协议标准设计测试例,最后在测试系统中执行测试例并对测试结果进行分析^[2]。

IPv6 一致性测试一直是国内外的研究热点,众多研究机构均已在这方面展开了研究工作,如美国新罕布什尔大学的 IOL 实验室^[3]、日本的 TAHI 计划^[4]等,Spirent 公司和 Ixia 公司也纷纷研制出了 IPv6 一致性测试仪;在国内,中科院计算技术研究所于 2001 年也展开了 IPv6 一致性测试的相关研究^[5,6],但其测试例

描述过于繁琐,对新协议的支持也不够。为提升我国在 IPv6 一致性测试领域的研究水平,文中经过深入分析和研究,提出了一致性测试的模块化测试描述模型,并自定义了相应的模块化测试描述语言(MTDL)来描述该模型。

1 模块化测试描述模型

从协议标准到可执行的测试例中间,需要有形式化的描述语言对协议标准进行解释,使由自然语言描述的协议标准转换成为可以被测试系统读懂并执行的测试例。参考 TTCN^[7]的结构框架和思路,结合 IPv6 协议的特点,提出了一种模块化测试描述模型。

模块化测试描述模型是指将 IPv6 协议栈中的每个协议作为一个对象,建立对应协议模块,协议模块包

收稿日期:2012-10-08

修回日期:2013-01-13

网络出版时间:2013-04-08

基金项目:国家自然科学基金资助项目(60903150,61100219);中央高校基本科研业务费专项资金(2012JBM010)

作者简介:张晓东(1989-),男,硕士研究生,研究方向为交换路由理论与技术;董平,博士,副教授,研究方向为新一代互联网、交换路由理论;苏伟,博士,副教授,研究方向为下一代互联网关键理论与技术。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20130408.1709.048.html>

括协议报头集和协议测试集。模块化测试描述模型如图 1 所示。

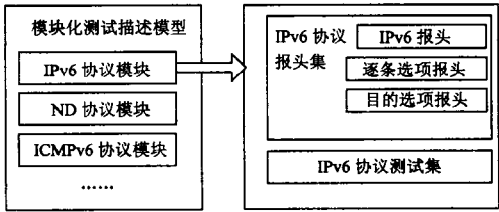


图 1 模块化测试描述模型

协议报头集包含了本协议的各个协议报头文件，如 IPv6 协议报头集中包含了 IPv6 协议首部和各个扩展首部的报头文件，在书写测试例的数据部分时，根据需从不同的协议报头集中取得各个协议报头有序组合即可，用户可对其中的字段做必要的修改。协议测试集包含了本协议的所有测试例，以脚本的形式保存在测试集中，测试例脚本的格式将在后面详细介绍。

模块化测试描述模型支持自定义测试例，用户可以根据自己的需要，使用模块化的协议报头集动态地进行测试例的增加、修改、删除操作。

1.1 模块化测试描述语言

采用自定义的模块化测试描述语言 (MTDL) 书写协议报头文件和测试例脚本文件。MTDL 按照功能分为数据描述和行为描述两部分。

MTDL 数据描述部分的语义规则如下：

(1) 协议报头名称和测试例名称以符号 “[” 和 “] ” 作为开始和结束符；文件说明部分以符号 “ # ” 作为开始和结束符；

(2) 关键字 “ Packet ” 和 “ Procedure ” 分别表示测试例的数据描述部分和行为描述部分，其内容以符号 “ { ” 和 “ } ” 作为开始和结束符；

(3) Packet 中的数据报由各个协议报头按顺序排列组成，在协议报头名称下方以 “ { ” 和 “ } ” 作为开始和结束符来表示需要修改的字段和参数；Packet 中可以包含一个或多个数据报，当包含多个数据报时，应以 Name 关键字给每个数据报命名；

(4) 所有数字赋值可以采用十进制数或采用十六进制数，十六进制数在数值前添加 “ 0x ” 前缀表示；数据 Data 赋值采用字符串格式，字符串内容以双引号作为开始和结束符。

MTDL 行为描述部分定义了多种测试行为，以下列出主要的行为名称及其参数：

(1) Empty: 表示清空测试系统的发送缓存，确保测试例在可靠的测试环境下进行，该行为命令没有参数；

(2) Send: 表示发送行为，有两个参数，一个参数表示要发送的数据包名称，例如 “ Packet_1 ”，另一个参数表示发送端口，如 “ Eth0 ”，每个参数均以符号 “ - ”

作为标识，参数没有先后顺序，整个发送行为命令的格式如 “ Send - Packet_1 - Eth0 ”；

(3) Receive: 表示监听测试系统端口并接收到达的数据包，有两个参数，一个参数表示接收端口，如 “ Eth1 ”，另一个参数表示超时时间，即 Receive 行为维持的最大时间，该参数以 “ Timeout ” 为关键字，其后跟数值表示超时时间，以毫秒为单位，关键字与数值之间以空格隔开，整个接收行为命令的格式如 “ Receive - Etn1 - Timeout 60 ”；

(4) Compare_Addr: 表示对接收到的数据包进行地址比较，有源地址和目的地址两个参数，分别以关键字 “ Src ” 和 “ Dst ” 表示，检验接收到的数据包的源地址和目的地址是否符合预期，若符合则进行下一步比较，若不符合则直接丢弃；

(5) Compare_Val: 表示对接收到的数据包的特定字段的值进行比较。该命令有三个参数，关键字 “ Offset ” 表示要进行比较的字段的偏移值（注意收到的数据包包含 MAC 首部），“ Value ” 表示要比较的对应偏移位置的值，“ Data ” 关键字表示要比较的是收到数据包的数据部分，其后跟数据包名称或者字符串，当是字符串时应该用双引号加以表示；

(6) Verdict: 根据 Compare_Addr 和 Compare_Val 的比较结果判定测试例是否通过，其后可跟文件名作为参数，表示将测试过程和结果输出到该文件中，不加文件名时，默认输出在屏幕上。

1.2 测试例描述方法

协议报头集包含了本协议的所有报头，下面以 IPv6 协议为例，介绍 IPv6 协议的报头集组成。

RFC2460^[8] 定义了以下扩展首部：逐跳选项首部、路由首部、分片首部和目的选项首部，并对扩展首部出现在数据报中的顺序提出了建议，模块化测试描述模型采用自定义的 MTDL 语言对 IPv6 协议^[9]的首部和扩展首部进行了描述，如表 1 所示。

表 1 IPv6 协议报头集

顺序	首部名称	MTDL 名称
1	IPv6 首部	IPv6_Header
2	逐跳选项首部	IPv6_HBH_Header
3	目的选项首部	IPv6_Dst_Header
4	路由首部	IPv6_Rte_Header
5	分片首部	IPv6_Frg_Header
-	选项首部	IPv6_Opt_Header

MTDL 对报头中的字段设置了默认值，以方便书写测试例脚本，如，IPv6_Header 报头的具体定义如下：

```
IPv6_Header
{
    Version= 0x06
```

```
Traffic_Class = 0x00
Flow_Label = 0x00000
Payload_Length = 0x0010
Next_Header = 0x3a
Hop_Limit = 0xff
Src_Addr = SOURCE_ADDR
Dst_Addr = DESTINATION_ADDR
}
```

SOURCE_ADDR 和 DESTINATION_ADDR 表示源和目的地址,存放在名为 ADDR_INFO.txt 的配置文件中,用户可直接打开该文件配置地址信息,也可通过测试系统提供的命令进行配置。

协议测试集包含了本协议的所有测试例的脚本文件,测试例脚本由 Packet 和 Procedure 两部分组成。采用协议报头组合树的方法对测试例脚本的 Packet 部分进行描述,按照测试例需求,将各个协议报头集的报头依次排列即可组成测试数据包,如果需要对某协议报头的字段值进行修改,只要在对对应协议报头下列出即可,这种方法大大简化了数据描述的工作量。Procedure 部分描述测试例的执行步骤,依次设置清空、发送、接收、比较、验证等行为命令及其参数值即可。下面举例说明如何应用 MTDL 语言书写测试例脚本。

IPv6 协议测试集中包含一个名称为 IPv6_OptionType_00 的测试例,测试目的是为了验证当 IUT 接收到包含有错误的 Option Type 值且其高两位为 00 的数据包时,应该忽略该错误并继续转发给目的地址。测试方法为由测试系统的两个端口模拟发送端 Tester A 和接收端 Tester B,被测设备 IUT 作为中间路由器,测试例脚本如下:

```
[IPv6_OptionType_00]
#Packet with an unrecognized Option Type and its #highest-order two bits set to 00.
Packet
{
    Name = Packet_OptionType_00
    {
        MAC_Header
        IPv6_Header
        {
            Next_Header = 0
        }
        IPv6_HBH_Header
        {
            IPv6_Opt_Header
            {
                Type = 0x0f
                Opt_Data_Length = 4
                Opt_Data = "1122"
            }
        }
    }
}
```

```
}
IPv6_Rqt_Header
DATA = "abcdefgh"
}
}
Procedure
{
    Empty
    Send -TesterA -Packet_OptionType_00
    Receive -TesterB -TimeOut = 50
    Compare_ADDR
    -Src_Addr SOURCE_ADDR
    -Dst_Addr DESTINATION_ADDR
    Compare_Val -Offset 62
    -Data "abcdefgh"
    Verdict Result.txt
}
```

(1) Packet 部分。

测试例脚本的 Packet 部分定义了一个名为 Packet_OptionType_00 的数据包,该数据包分别由 MAC_Header、IPv6_Header、IPv6_HBH_Header、IPv6_Opt_Header 以及 IPv6_Rqt_Header 和 DATA 报头组成。按照测试例要求,将 IPv6_Header 报头中的 Next_Header 字段值设置为 0,表明下一个首部是逐跳选项首部;将 IPv6_Opt_Header 报头中的 Type 字段修改成 0x0f(二进制表示为 00001111,高两位为 00 的错误值),选项长度字段和数据字段也做相应修改,最后为数据包加上内容为“abcdefgh”的数据,组数据包的工作完成。

(2) Procedure 部分。

Procedure 部分将 Send 命令的发送端口设置为 TesterA,发送数据包为 Packet 部分定义的数据包 Packet_OptionType_00;将 Receive 命令的接收端口设置为 TesterB,超时时间设置为 50ms;将 Compare_Addr 命令的源、目的地址参数分别设置为 TesterA 和 TesterB 的 IP 地址;Compare_Val 命令设置了一组参数,比较收到数据包的数据部分,偏移值为 62,比较数据即为 TesterA 端发送的数据包 Packet_OptionType_00 的数据部分;最后使用 Verdict 命令将测试过程和结果输出在名为 Result.txt 的文件中。

2 基于 MTDL 的一致性测试系统

按照模块化测试描述模型的设计思想,文中设计和开发了基于 MTDL 的 IPv6 协议一致性测试系统^[10,11],测试系统由三部分组成:协议报头调度器、测试例编译器和测试例执行器^[12]。测试系统的功能结构如图 2 所示。

协议报头调度器负责从各协议报头集中调取报头信息,传递给编译器的 Packet 编译模块;编译器的

Packet 编译模块和 Procedure 编译模块分别编译测试脚本中的 Packet 和 Procedure 部分,并将编译好的数据传递给执行器的 Packet 数据中心和 Procedure 执行中心;执行器根据执行中心存储的行为命令,调取 Packet 数据执行测试例,并生成测试报告。

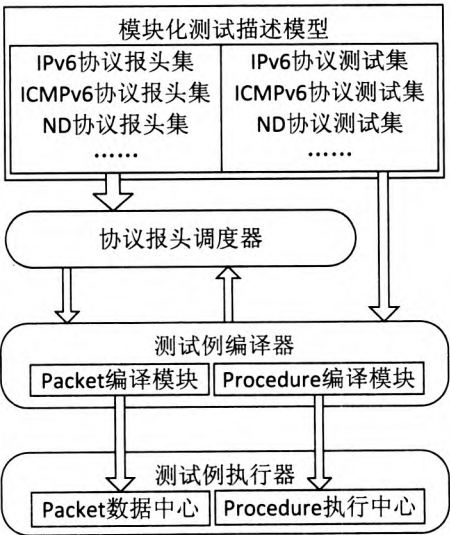


图 2 基于 MTDL 的一致性测试系统

3 IPv6 协议一致性测试

文中在对 IPv6 协议规范(RFC2460)的内容进行分析的基础上,设计了基于 MTDL 的 IPv6 协议测试集,并在 Linux 平台下对一种 IPv6 实现进行了测试,给出了测试结果。

3.1 测试例设计

根据 RFC2460 的规范,文中设计了五个测试组共 37 个测试例,如表 2 所示。

表 2 IPv6 协议测试例

测试组名称	测试组描述
IPv6 首部测试组	针对首部中流量类别、流标签、有效载荷长度、下一个首部和跳数限制五个字段设计了 5 个测试例
选项首部测试组	针对逐跳选项首部和目的选项首部中 Option 选项设计了 12 个测试例
路由首部测试组	针对路由首部中路由类型、扩展首部长度、剩余段数以及类型特有数据等字段设计了 11 个测试例
分片首部测试组	针对分片首部设计了 7 个测试例
多个扩展首部测试组	针对多个扩展首部时的处理规则设计了 2 个测试例

3.2 测试实现及结果分析

文中选取的被测设备(IUT)为一台装有 Linux 操作系统的路由设备,系统版本为 Fedora 10,内核版本为 2.6.27.5-117。测试设备(Tester)实现了 IPv6 协议一致性测试系统,由于 IUT 只根据接收到的数据包

的 IP 地址来判断数据发送方身份,因此,文中采用了测试设备的两个端口模拟两台不同的设备 Tester A 和 Tester B,测试拓扑如图 3 所示。

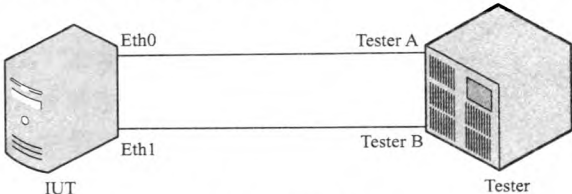


图 3 测试拓扑

下面对文中第 1.2 小节提到的名称为 IPv6_OptionType_00 的测试例进行测试。测试例执行过程如图 4 所示,Tester A 向 Tester B 发送一个名为 Packet_OptionType_00 的数据包,从图中可以看出,Tester B 在某时刻接收到了一个数据包,并对其地址和关键值进行了验证,结论均为 Correct,由此可以判定此数据包正是 Tester A 发送的数据包,最终测试例执行结果为 PASS。

```
*****
Test Begin 2012-9-28 21:31:37.451639
-----
Empty 2012-9-28 21:31:37.455384
-----
Tester A SEND Packet 2012-9-28 21:31:38.454723

Packet Name : Packet_OptionType_00
Length : 78 Bytes
Data :
40 61 86 51 5b 59 00 24 21 3d 6a d4 86 dd 60 00
00 00 00 18 00 ff 3f fe 0c 00 00 00 00 01 02 60
00 10 94 00 03 a8 3f fe 0c 00 00 00 00 01 02 60
00 06 28 d8 fe 60 3a 00 0f 04 31 31 32 32 80 00
92 04 00 64 00 01 61 62 63 64 65 66 67 68
-----
Tester B RECEIVE Packet 2012-9-28 21:31:38.455194

Length : 78 Bytes
Data :
00 24 21 3d 6a d4 40 61 86 51 5b 59 86 dd 60 00
00 00 00 10 3a 40 3f fe 0c 00 00 00 00 01 02 60
00 06 28 d8 fe 60 3f fe 0c 00 00 00 00 01 02 60
00 10 94 00 03 a8 81 00 91 04 00 64 00 01 61 62
63 64 65 66 67 68
-----
Compare_ADDR 2012-9-28 21:31:38.456403
Correct
-----
Compare_Val 2012-9-28 21:31:38.456633
Correct
-----
Verdict 2012-9-28 21:31:38.456787
Compare_ADDR : OK Compare_Val : OK
Conclusion : Test Case IPv6_OptionType_00 PASS!
*****
```

图 4 IPv6_OptionType_00 测试过程

文中设计的 37 个测试例经过在 IUT 上实际测试,全部通过,说明 IUT 很好地实现了 IPv6 协议标准。

4 结束语

文中提出了 IPv6 协议一致性测试的模块化测试描述模型,设计了一种模块化测试描述语言 MTDL,对该语言进行了详细的描述,并开发了一套基于该语言的 IPv6 协议一致性测试系统。文章通过对 IPv6 协议

(下转第 47 页)

可得三次实验的各个参数值和结果如表 1 所示:

表 1 使用语义合成技术前后的查全率和查准率

s	sp	w1	wp1	w2	wp2	r1	r2	p1	p2
100	50	60	48	40	35	0.96	0.7	0.8	0.875
1000	100	200	100	110	98	1.0	0.98	0.5	0.89
2000	200	800	198	210	190	0.99	0.95	0.25	0.9

从表中可看出,通过比较一般检索和使用语义合成技术后的检索的查全率和查准率,可得出如下分析结果:使用了语义合成技术后的查全率稍稍低于普通检索,但查准率高于不使用语义合成技术的普通检索,且查询效率明显高于普通的检索;并发现随着检索系统中的总文献数的增加,基于语义合成技术的检索优点就会显得越突出。这是因为通过语义合成技术,过滤了一些语义无关的信息,从而提高了检索质量。

5 结束语

文中提出了基于领域本体的语义合成的研究,将待查询信息与本体技术相结合,有效地解决了查询过程中的语义问题;本体为用户输入的查询描述加入了语义信息,提高了信息查询的准确度和智能化,提高了查询的质量和效率,从而使计算机处理文本信息的能力和品质也改善了,与此同时,用户信息检索的满意度也提高了。

文中下一步可以针对不同的数据源或数据对象进行语义合成,将该语义合成技术进一步深入研究,并应用到更广阔的领域。

参考文献:

[1] 刘 琼,李宝敏.一种果品领域本体库的构建方法[J]. 计算机技术与发展,2009,19(1):197-199.

[2] Gruber T R. Towards Principles for the Design of Ontologies Used for Knowledge Sharing[J]. International Journal of Human Computer Studies,1995,43(5/6):907-928.

[3] Studer R,Benjamins V R,Fensel D. Knowledge Engineering, Principles and Methods[J]. Data and Knowledge Engineering,1998,25(1-2):161-197.

[4] 王 宁,王能斌.异构数据源集成系统查询分解和优化的实现[J]. 软件学报,2000,11(2):222-228.

[5] Tomasic A,Amouroux R,Bonnet P,et al. The distributed information search component (Disco) and the World Wide Web[J]. ACM SIGMOD Record,1997,26(2):546-548.

[6] 王昭龙,李 霞,许瑞芳.多关键字查询中 LCA 剪枝概念树的查询扩展技术研究[J]. 计算机科学,2010,37(4):132-135.

[7] 赵 心.一种基于关联规则的中午概念集生产算法[J]. 计算机科学,2004,31(7):175-177.

[8] 韩家炜,孟小峰,王 静,等. Web 挖掘研究[J]. 计算机研究与发展,2000,37(5):513-520.

[9] 吴 建,吴朝晖,李 莹,等.基于本体论和词汇语义相似度的 Web 服务发现[J]. 计算机学报,2005,28(4):595-602.

[10] Navigli R,Velardi P. Learning domain ontologies from document warehouses and dedicated web sites[J]. Computational Linguistics,2004,30(2):151-179.

[11] 黄 果,周竹荣.基于领域本体的概念语义相似度计算研究[J]. 计算机工程与设计,2007,28(10):2460-2463.

(上接第 43 页)

标准的分析,设计了一套 IPv6 协议测试集,利用自主开发的测试系统对 Linux 下的一种 IPv6 协议实现进行了测试,给出了测试结果,并对结果进行了分析,验证了该协议实现的正确性和完备性。目前,基于模块化测试描述模型和 MTDL 的协议测试集还不够完备,后续将继续致力于协议测试例的开发,提高可测试协议的覆盖度,并进一步完善测试系统功能。

参考文献:

[1] 田 军,张玉军,李忠诚. IPv6 协议一致性测试系统[J]. 计算机辅助设计与图形学学报,2002,14(4):296-300.

[2] 郑红霞,田 军,张玉军,等. IPv6 协议一致性测试例的设计[J]. 计算机应用,2003,23(4):62-64.

[3] Interoperability Lab:IP Consortium Test Suite, Internet Protocol Version 6[M]. Hampshire:University of New Hampshire, 2000.

[4] Tahi project. IPv6 node requirements revision[EB/OL]. 2004-10. <http://www.tahi.org/ume/testspec/ph1-host-policy.html>.

[5] Tian Jun,Li Zhongcheng. The next generation Internet protocol and its test[C]//Proc. of IEEE International Conference on Communications. [s. l.]:[s. n.],2001:210-215.

[6] Zhang Yujun,Li Zhongcheng. A new formal test suite specification language for IPv6 conformance testing[C]//Proceedings of International Conference on Communication Technology. [s. l.]:[s. n.],2003:174-177.

[7] ISO/IEC 9646:IT-OSI-Conformance testing methodology and framework[S]. [s. l.]:[s. n.],1996.

[8] RFC2460: Internet Protocol, Version 6 (IPv6) Specification [S]. [s. l.]:Network Working Group,1998.

[9] Li Qing,Jinmei T,Shima K. IPv6 详解[M]. 北京:人民邮电出版社,2009.

[10] 张 升,刘兴伟,郭 闯. IPv6 协议一致性测试[J]. 西华大学学报:自然科学版,2009,28(1):34-37.

[11] 夏启志. IPv6 协议一致性测试通用执行系统设计与实现[D]. 北京:中国科学院计算技术研究所,2005.

[12] 孙静波,张玉军,李忠诚. IPv6 中邻居发现协议及其测试[J]. 计算机工程与应用,2004,24(32):79-81.

IPv6协议一致性测试描述语言设计与实现

作者: 张晓东, 董平, 苏伟, ZHANG Xiao-dong, DONG Ping, SU Wei
作者单位: 北京交通大学电子信息工程学院, 北京, 100044
刊名: 计算机技术与发展 
英文刊名: Computer Technology and Development
年, 卷(期): 2013, 23(7)

本文链接: http://d.wanfangdata.com.cn/Periodical_wjfz201307010.aspx