

IPv6 邻居发现协议的一致性测试序列生成

陈守宁¹ 郑宝玉¹ 李 璟² 赵玉娟¹

(1. 南京邮电大学 通信与信息工程学院, 江苏 南京 210003;

2. 江苏移动南京分公司, 江苏 南京 210029)

摘 要: 自 1998 年互联网工程任务组(IETF)提出下一代互联网标准规范以来, IPv6 已经历了十多年的发展。现今已有越来越多的 IPv6 产品被投入到了开发与应用中。而如何提高不同产品间的互通性和可靠性则成为了一个关键问题。进行协议一致性测试是提高 IPv6 实现可靠性的一种有效方式。本文就重点针对 IPv6 邻居发现协议进行了一致性测试分析。本文首先简要分析了 IPv6 邻居发现协议的主要功能及实现原理, 并据此抽象出其有限状态机(FSM)模型。进而结合一种现有基于有限状态机(FSM)的一致性测试序列改进算法生成了该协议的抽象测试序列。本文在最后对得到的测试序列进行了有效性和可靠性分析, 分析表明, 使用该算法得到的测试序列不仅在序列长度上较传统 UIO 序列法有了明显的缩短, 同时对测试过程中可能发生的输出错误及末状态转换错误也具备良好的检测能力。本文获得的抽象测试序列可对相关 IPv6 协议开发者提供有效参考。

关键词: IPv6; 协议一致性测试; 有限状态机(FSM); UIO 序列

中图分类号: TN92 **文献标识码:** A **文章编号:** 1003-0530(2013)12-1670-07

Generating of Conformance Testing Sequence for IPv6 Neighbor Discovery Protocol

CHEN Shou-ning¹ ZHENG Bao-yu¹ LI Jing² ZHAO Yu-juan¹

(1. College of Telecommunications & Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China; 2. Jiangsu Mobile Communication Company Limited Nanjing Branch, Nanjing 210029, China)

Abstract: Since the Internet Engineering Task Force (IETF) put forward the next generation of Internet standards in 1998, IPv6 (Internet Protocol version 6) has gone through ten years of evolution and nowadays more and more IPv6 devices have been put into development and application. So how to improve the reliability and interoperability between different IPv6 implementations has become a key point. Conformance testing is an effective method to improve the reliability of IPv6 implementation and this paper will focus on the conformance testing of IPv6 Neighbor Discovery Protocol (NDP). In this paper we first analyze the major function and principle of NDP and present the finite-state machine (FSM) model. Based on an existing improved algorithm for conformance testing we derive the abstract conformance testing sequence for NDP. We also analyse the validity and reliability of this algorithm. The analysis results show that the length of the testing sequence obtained by using this algorithm is shorter than traditional U-method and the detection coverage for output error and tail state error is favourable. The abstract conformance testing sequence we derived in this paper can provide meaningful reference to IPv6 protocol developers.

Key words: IPv6; conformance testing; finite-state machine (FSM); UIO sequence

1 引言

IP 网络是以 TCP/IP 协议为基础通信协议的网络,著名的 Internet 就是最具代表性的 IP 网络。但是,随着网络规模的不断扩大,IPv4 的缺陷和不足,如地址资源紧张;网络号匮乏;路由表急剧膨胀以及不能很好地支持实时业务等问题使得它正逐渐不能适应高速发展的 Internet 的要求。为此,IETF 制定了用以取代 IPv4 的新一代 Internet 协议—IPv6。IPv6 协议在 IPv4 的基础上,根据 IPv4 多年来运行的经验进行了大幅度的修改和功能扩充^[1]。

由于目前的协议标准基本上都是使用自然语言描述的,包括 IPv6 协议。实现者对于协议的不同理解会导致不同的协议实现,甚至有时会是错误的实现,因此我们需要一种有效的方法来对协议实现的正确性和有效性进行判别,这便是“协议测试”^{[2][3]}。

协议测试包含三种类型的测试:一致性测试、互操作性测试和性能测试。其中一致性测试是依据一个协议的描述文本对协议的某个实现进行测试,是通过观察具体实现在不同的环境和条件下的响应来验证协议的实现与相应的协议标准是否一致,用以保证不同网络产品的协议实现之间能够进行有效可靠的互联互通。所以一致性测试是协议测试的最基本内容,是其他测试的基础^[4]。

一致性测试属于“黑盒测试”,即将被测系统视为一个黑盒,不关心其程序的内部结构,仅仅通过观察外部行为来判断软件功能实现的正确性。形式化描述技术是将由自然语言描述的协议抽象为具有严密逻辑结构的形式描述语言,也称之为“建模”。使用形式化描述语言来描述协议,编写测试用例,进行系统仿真,通过控制观察点观察被测实现(IUT, Implement Under Test)对激励信号的响应来验证协议实现与协议规范的一致性。本文依据 IPv6 邻居发现协议(NDP, Neighbor Discovery Protocol)的特性,采用有限状态机(FSM)对其进行建模并生成了抽象一致性测试序列。应用开发者或产品开发者可有效地利用该方法对其 IPv6 产品进行互联互通测试。

本文第 2 节简要介绍了 IPv6 邻居发现协议的基本功能及实现原理,并具体分析了邻居发现协议的 FSM 模型;在第 3 节我们结合唯一输入/输出序列(Unique Input/Output sequences)即 UIO 序列以及周游法,通过一种改进算法得到了邻居发现协议的抽象一致性测试序列;文章第 4 节对通过该算法获得的抽象测试序列进行了有效性和可靠性分析;在最后一节中对全文进行了总结。

2 IPv6 邻居发现协议(NDP)及其有限状态机(FSM)模型

2.1 NDP 介绍

NDP^[5]是 IPv6 协议簇的一个重要组成部分,主要解决了同一链路上不同节点间的信息交互问题。它实现了 IPv4 中地址解析协议、控制报文协议(ICMP)中路由发现协议和重定向协议的所有功能,并在它们的基础上进行了相应的修改和功能扩充。

具体地讲,NDP 通过使用 5 种类型的 IPv6 控制消息报文(ICMPv6)实现了其 9 大功能。其中 5 种消息报文分别为:路由请求、路由宣告、邻居请求、邻居宣告以及重定向消息。9 大功能分别为:路由器发现、前缀发现、参数发现、地址自动配置、地址解析、下一跳地址确定、邻居不可达检测、重复地址检测和重定向功能^[6]。

2.2 NDP 的 FSM 模型

对于抽象测试序列的生成我们可以先利用一般的“建模”方式生成形式化模型。根据不同协议的特征可以选用不同的建模方法。比如基于 NDP 中邻居设备所处的不同状态,我们选择使用有限状态机(FSM)为其进行建模。

带标记的变迁转换系统以 FSM 模型为基础,通过状态之间的变迁得到形式化模型,生成抽象测试用例^[7]。带标记的变迁转换系统可以定义为一个六元组: $\{\Sigma, \Gamma, s_1, S, \delta, \omega\}$, 其中: $\Sigma = \{\sigma_1, \sigma_2, \dots, \sigma_k\}$ 表示有限输入字符集; $\Gamma = \{\gamma_1, \gamma_2, \dots, \gamma_m\}$ 表示有限输出字符集; $S = \{s_1, s_2, \dots, s_n\}$ 表示有限状态集; $s_1 \in S$ 为初始状态; $\delta: S \times \Sigma \rightarrow S$ 为状态转移函数;

$\omega: S \times \Sigma \rightarrow \Gamma$ 为输出函数。协议的 FSM 模型通过有向图 $G=(V,E)$ 表示,其中 $V=\{v_1, v_2, \dots, v_m\}$ 是有向图的顶点集合,等效于 FSM 的状态 S ; $E=\{e_1, e_2, \dots, e_m\}$ 是有向边集合,其中 $e_{ij}=(v_i, v_j, i/o)$ 表示从状态 s_i 到状态 s_j 的转移; i/o 等效于 FSM 中的输入输出字符。以 NDP 为例的 FSM 模型如图 1 所示。

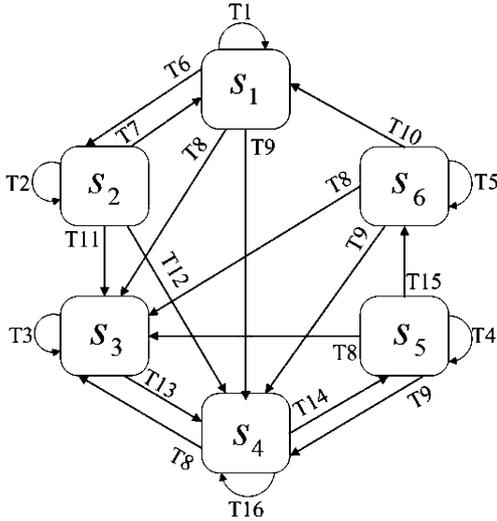


图 1 G:邻居发现协议的有限状态机模型

图 1 G: FSM Model of NDP

其中有限状态集 $S=\{s_1, s_2, s_3, s_4, s_5, s_6\}$ 分别表示空闲、未完成、可达、失效、延迟、探测这六个邻居状态。空闲状态表示节点还未建立相关邻居的缓存表项;未完成状态表示正在解析邻居的链路层地址;可达状态表示地址解析成功,该邻居可达;失效状态表示可达时间耗尽,不确定邻居是否依旧可达;延迟状态同样表示不确定邻居是否可达,是一个延时等待状态;节点会向处于探测状态的邻居持续发送邻居请求报文。若在规定时间内一直未收到其回应的宣告报文,则删除该邻居的缓存表项,回到空闲状态。

图 1 中 16 条有向边分别表示 16 个测试事件集,每个事件集分别包含若干个输入输出(i/o)事件。其中 $T1=\{9,10\}$; $T2=\{1\}$; $T3=\{5,8,9,10,12\}$; $T4=\{7,9,10\}$; $T5=\{7,9,10,16\}$; $T6=\{0\}$; $T7=\{2\}$; $T8=\{5,8,12\}$; $T9=\{11\}$; $T10=\{17\}$; $T11=\{4\}$; $T12=\{3\}$; $T13=\{6,11,13\}$; $T14=\{14\}$; $T15=\{15\}$; $T16=$

$\{7,9,10,11\}$,各事件的 i/o 如表 1 所示:

表 1 邻居发现协议的 i/o 事件集

表 1 i/o Events of NDP

i		o
0	希望向对方节点发包	建立对方节点邻居缓存表项,发送多播邻居请求(Neighbor Solicitation, NS),开启重传计时器
1	重传计时器超时,且计时器重启次数不超过 N 次	重启计时器,同时发送 NS
2	计时器超时,且重启次数大于或等于 N	丢弃报文,返回 ICMP 错误消息
3	收到非请求的邻居宣告(Neighbor Advertisement)	记录邻居链路层地址,发送队列中的包
4	收到请求的 NA	记录邻居链路层地址,发送队列中的包
5	收到请求的 NA,与缓存中的链路地址相同,不要求覆盖	—
6	收到请求的 NA,与缓存中的链路地址不同,不要求覆盖	—
7	收到请求的 NA,与缓存中的链路地址不同,不要求覆盖	—
8	收到请求的 NA,要求覆盖	当与缓存中的链路地址不同时记录新地址
9	收到非请求的 NA,不要求覆盖	—
10	收到非请求的 NA,与缓存中的链路地址相同,要求覆盖	—
11	收到非请求的 NA,与缓存中的链路地址不同,要求覆盖	记录新地址
12	收到上层可达性确认	—
13	超时,自上次可达性确认后已超过 N 秒	—
14	希望向对方节点发包	启动延时计时器
15	延时超时	发送可达性单播 NS 请求探测,启动重传计时器
16	重传计时器超时,且计时器重启次数不超过 N 次	重启计时器,同时发送 NS
17	计时器超时,且重启次数大于或等于 N	删除邻居缓存表项

需要注意的是, 接到除邻居宣告信息外非请求消息的情况在这里不做讨论。

3 NDP 抽象一致性测试序列的生成

采用第 1 节中 FSM 模型的目的是希望在图 1 中找到一条能够遍历所有状态迁移的路径(欧拉路径), 把这个路径上的输入输出前后连接起来便生成了可以用来测试的输入输出序列, 即抽象测试序列。但是在这一过程中需要解决一些问题, 例如: 如何判断 G 是否为欧拉图; 若不是如何添加最少的冗余边将 G 构造为欧拉图; 如何判断协议所处的状态; 测试过程中发现错误后如何根据测试序列准确定位错误所在位置等。在实际操作过程中可以通过不同的方法解决上述问题, 如周游法、区分序列法、特征序列法、UIO 序列法^[8]。在这里我们使用一种对 UIO 序列优化的方法, 即结合周游方法来产生最小测试代价的测试序列。在实际测试中使用这种方法与使用 UIO 方法的错误检测能力相同, 但产生的测试序列却比 UIO 方法短很多^[9]。本文 NDP 抽象测试序列的自动生成就是采用了该优化算法, 具体步骤如下:

①根据 NDP 的 FSM 图, 计算每个状态的 UIO 序列及 UIO(s_i) 之后的末状态 s_j (表 2)。

②找出图 1 中 G 的汇聚边(若状态 s_m, s_n 经过同样的 i/o 转换到相同的状态 s_t , 则 $(s_m, s_t; i/o)$, $(s_n, s_t; i/o)$ 称为汇聚边)。 G 有七条汇聚边, 分别为: $(s_1, s_3; T8)$; $(s_4, s_3; T8)$; $(s_5, s_3; T8)$; $(s_6, s_3; T8)$ 及 $(s_1, s_4; T9)$; $(s_5, s_4; T9)$; $(s_6, s_4; T9)$ 。

表 2 G 的 UIO 序列
表 2 UIO Sequence of G

s_i	UIO	s_j
s_1	$T6$	s_2
s_2	$T11$	s_3
s_3	$T13$	s_4
s_4	$T14$	s_5
s_5	$T15$	s_6
s_6	$T10$	s_1

③对每一个以汇聚边的初始状态为末状态的 i/o 序列, 计算其测试子序列 $e(s_i, s_j, s_i; i/o) = \{i/o, UIO(s_j)\}$, 即:

$$e1 = (s_1, s_1, s_2, T1) = \{T1, T6\}; e7 = (s_2, s_1, s_2, T7) = \{T7, T6\}; e10 = (s_6, s_1, s_2, T10) = \{T10, T6\}。$$

$$e9 = (s_1, s_4, s_5, T9) = \{T9, T14\}; e12 = (s_2, s_4, s_5, T12) = \{T12, T14\}; e13 = (s_3, s_4, s_5, T13) = \{T13, T14\}; e16 = (s_4, s_4, s_5, T16) = \{T16, T14\}; e9 = (s_5, s_4, s_5, T9) = \{T9, T14\}; e9 = (s_6, s_4, s_5, T9) = \{T9, T14\}。$$

$$e14 = (s_4, s_5, s_6, T14) = \{T14, T15\}; e4 = (s_5, s_5, s_6, T4) = \{T4, T15\}。 e15 = (s_5, s_6, s_1, T15) = \{T15, T10\}; e5 = (s_6, s_6, s_1, T5) = \{T5, T10\}。$$

④构造有向图 G' , 用 $E = (e_1, e_4, e_5, \dots, e_{15})$ 代替 G 中的 i/o , 如图 2 所示。

⑤采用 Dijkstra 算法得到各状态间的最短路径, 在非平衡状态 s_i, s_j 间添加取自于 G 的弧, 将 G' 转化为最优有向对称图 G^* , 如图 3 所示。

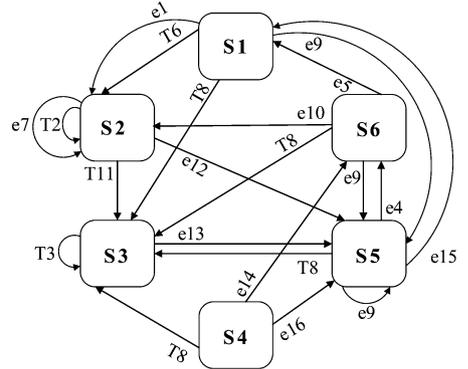


图 2 非平衡图 G'

图 2 G' : unbalanced graph

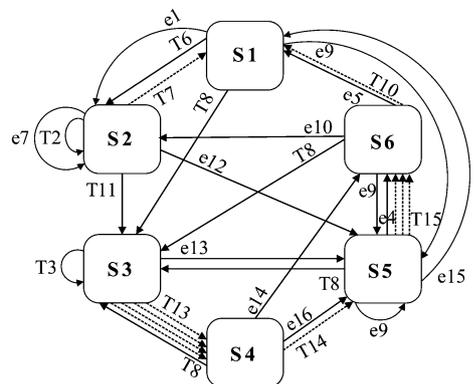


图 3 有向对称图 G^*

图 3 G^* : directed symmetric graph

⑥ 构造 G^* 的邻接矩阵,以初始状态 s_1 为起点历经有向边 E_{1k} (k 为有向边的末状态)。再以 k 为起点历经下一条边。以此类推,直到遍历 G^* 的所有有向边即可获得欧拉游程 (EP)。遍历序列如下所示:

EP: $s_1 \xrightarrow{T6} s_2 \xrightarrow{T7} s_1 \xrightarrow{e1} s_2 \xrightarrow{e7} s_2 \xrightarrow{T2} s_2 \xrightarrow{T11} s_3 \xrightarrow{T3} s_3 \xrightarrow{T13} s_4 \xrightarrow{T8} s_3 \xrightarrow{T13} s_4 \xrightarrow{e16} s_5 \xrightarrow{e15} s_1 \xrightarrow{T8} s_3 \xrightarrow{T13} s_4 \xrightarrow{T14} s_5 \xrightarrow{T8} s_3 \xrightarrow{T13} s_4 \xrightarrow{e14} s_6 \xrightarrow{e5} s_1 \xrightarrow{e9} s_5 \xrightarrow{e9} s_5 \xrightarrow{e4} s_6 \xrightarrow{e10} s_2 \xrightarrow{e12} s_5 \xrightarrow{T15} s_6 \xrightarrow{T8} s_3 \xrightarrow{e13} s_5 \xrightarrow{T15} s_6 \xrightarrow{e9} s_5 \xrightarrow{T15} s_6 \xrightarrow{T10} s_1$ 。在以上遍历过程中,用基本状态转移有向边替换测试子序列便可以得到所需的测试序列。

4 有效性与可靠性分析

4.1 有效性分析

① 计算每个状态的最短 UIO 序列复杂度为 $O(n^2(d_{max})^{2n^2+2})$,其中 n 为图 1 中 G 的状态数, d_{max} 为各个状态的最大出度;

② 对于汇聚边,利用数据结构的知识,将 G 中所有有向边按照其对应的输入输出对来分类,分别插入到对应的链表中。将各个链表中每条有向边的末状态进行比较,若相同则为汇聚边。所有链表的长度都不会超过有向边总数 m ,比较链表中各元素的复杂度相当于给链表进行排序,为 $O(m * \log(m))^{[10]}$ 。即便所有的输入输出对都不同,那链表数最多为 m 个。所以这一步的总复杂度为 $O(m^2 \log(m))$;

③ 汇聚边不会超过总的有向边数 m ,所以其测试子序列的时间复杂度为 $O(m)$;

④ 用测试子序列代替原输入输出对的时间复杂度也为 $O(m)$;

⑤ 采用 Dijkstra 算法得到各状态间的最短路径,在非平衡状态 s_i, s_j 间添加取自于 G 的弧,使得它们的出入度之差依次减 1。Dijkstra 算法的复杂度为 $O(n^2)$;

⑥ 求图 3 中 G^* 的欧拉游程。因为添加的冗余弧都来自于 G ,而每个状态的不平衡度小于 m ,所以

G^* 中的有向边总数不会超过 $m+m^2$,欧拉游程的复杂度为 $O(m^2)$ 。

将以上各步叠加,得到算法总复杂度为 $O(n^2(d_{max})^{2n^2+2})+O(m^2 \log(m))$ 。

4.2 可靠性分析

这里的可靠性分析即是对协议抽象测试序列错误覆盖率的分析。我们可以根据协议的 FSM 建立错误的方程化模型,对不同的错误类型分别对其覆盖率进行量化^[11]。错误类型大致可分为两类:

- 1:若 FSM 中某有向边 E 的输出发生了错误,则称 E 发生了类型 1 的错误;
- 2:若该有向边 E 的末状态由 i 错误地指向了 j ,则称 E 发生了类型 2 的错误,即 $TAIL(E) = i$, $FAULTYTAIL(E) = j$;

对于完全确定的 FSM,任何类型 1 的错误都可以通过 UIO 检测出来,因为根据其性质,任何状态的 UIO 输入/输出序列都是不同的。所以 UIO 序列对错误类型 1 的检测率是 100%的。

对于类型 2 的错误,我们也可以使用 E 的末状态 i 的 UIO $_i$ 序列来检测,图 4。

$E = (v_r, v_i, i/o)$ 表示正确的状态转移, $E = (v_r, v_i, i'/o)$ 表示 E 发生了类型 2 的错误。即 $TAIL(E) = i$, $FAULTYTAIL(E) = i'$ 。在末状态 i/i' 后面分别添加 i 的 UIO 序列,对于左右两条状态转移流程图,同一水平线上的输入相同,根据 UIO 序列的性质,若输出不同则说明发生了类型 2 的错误。但在某些情况下也有可能出现如图 5 的错误。这种错误仅仅依靠 UIO 序列就不能被检测出来。

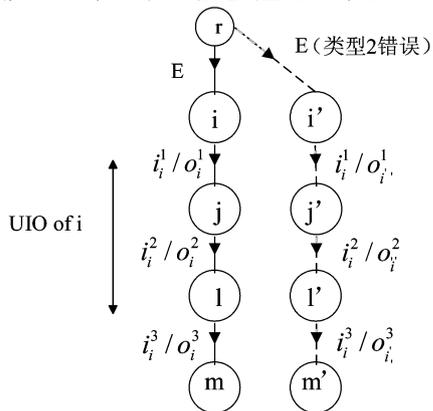


图 4 可被检测到的类型 2 错误

图 4 Type 2 fault which can be detected

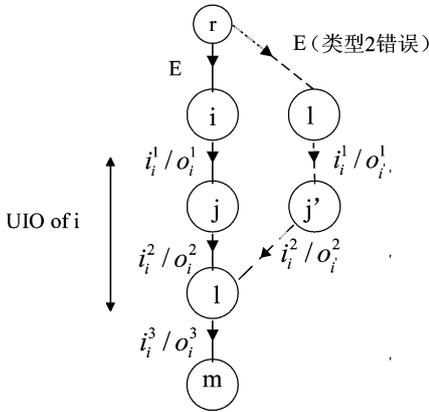


图 5 不可被检测到的类型 2 错误

图 5 Type 2 fault which can not be detected

通过观察我们可以发现,必须同时具备如下两个条件才有可能发生图 5 的错误:

1、E 的输入输出对与 i 的 UIO 序列的第 n 条边 UIOⁿ的输入输出对相同,比如图 5 中的 UIO²(i_i^2/o_i^2);

2、E 中发生错误的末状态与 UIOⁿ的末状态相同,即 FAULTYTAIL (E) = TAIL (UIO²),如图 5 中的 l。

具体分析 NDP 的 FSM,可能发生类型 2 错误的有向边共有 95 条,但没有一条同时具备以上两个条件的。所以,UIO 序列对 NDP 抽象测试序列中可能发生的类型 2 错误的检测覆盖率也是 100% 的。

需要补充说明的是,根据唯一输入输出序列的性质,任何状态的 UIO 输入/输出序列都是不同的。我们是通过唯一的输出结果判断测试序列的转换是否发生错误,所以不存在虚警概率。表 3 给出了 IPv6 协议在不同测试序列算法下的有效性及可靠性对比。

表 3 有效性及可靠性对比

表 3 Comparison of validity and reliability

	测试序列长度	错误检测覆盖率
周游法	41	无法检测
UIO 序列法	54	100%
本文优化算法	44	100%

5 测试验证

本文使用形式化描述语言 SDL 进行仿真。测

试系统模型如图 6 所示,其中被测实现即为 IPv6 协议,下测试器提供测试激励同时给出测试判决。

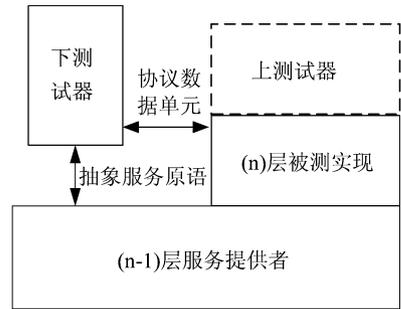


图 6 测试系统模型

图 6 Test system model

使用以上测试系统及序列对 IPv6 协议进行了测试,测试事件状态表如表 4 所示。测试结果显示序列转换顺序正确,未出现任何输出或未状态转换错误。

表 4 测试事件状态表

表 4 Test event state table

测试事件	T ₆	T ₇	T ₁₀
初始状态	S1	S2	S6
i	准备发包	计时器超时	计时器超时
末状态	S2	S1	S1
o	NS	ICMP	删除邻居缓存表
测试结果	Pass	Pass	Pass

6 结束语

本文通过分析 IPv6 邻居发现协议的基本功能及实现原理,选择使用有限状态机(FSM)来生成其形式化模型。并通过一种结合 UIO 序列以及周游法的一致性测试序列改进算法得到了邻居发现协议的抽象测试序列。最终对得到的测试序列进行了有效性和可靠性分析,验证了所用算法可以完全覆盖对邻居发现协议进行测试时可能出现的所有错误类型。应用开发者或产品开发者可利用该方法对其 IPv6 产品进行有效的互联互通测试。

参考文献

- [1] 徐宇杰. IPv6 深入分析[M]. 清华大学出版社, 2009.
XUN Y J. IPv6 in-depth analysis[M]. Tsinghua University Press, 2009. (in Chinese)
- [2] Zhang Y, Li Z. IPv6 conformance testing: theory and practice[C]//Test Conference, 2004. Proceedings. ITC 2004. International. IEEE, 2004: 719-727.
- [3] 李忠诚, 张玉军. IPv6 协议一致性测试的实现[J]. 电信技术, 2003, 7: 009.
LI Z C, ZHANG Y J. The realization of conformance testing to IPv6 protocols[J]. Telecommunications Technology, 2003,7:009. (in Chinese)
- [4] Vallejo A, Ruiz J, Abella J, Zaballos A, Selga J M. State of the art of IPv6 conformance and interoperability testing [J]. Communications Magazine, IEEE, 2007, 45(10): 140-146.
- [5] Narten T, Simpson W A, Nordmark E, Soliman H. Neighbor discovery for IP version 6 (IPv6)[J]. RFC 4861, 2007.
- [6] 李伟, 唐伦, 陈前斌. 基于 IPv6 的邻居发现协议性能分析[J]. 中国新通信, 2006, 8(21):54-59.
LI W, Tang L, CHEN Q B. Performance analysis of neighbor discovery protocol based on IPv6[J]. China New Telecommunications, 2006, 8(21):54-59. (in Chinese)
- [7] 叶新铭, 孙美飞. IPv6 邻居发现协议的一致性测试 [J]. 计算机科学, 2005, 32(6): 43-46.
YE X M, SUN M F. The conformance testing of neighbor discovery for IPv6 [J]. Computer Science, 2005, 32(6):43-46. (in Chinese)
- [8] 袁博, 李进, 孙海波, 杨盘隆. 协议测试中测试序列生成方法综述[J]. 军事通信技术, 2008, 29(1): 50-52.
YUAN B, LI J, SUN H B, YANG P L. Review of test sequence generation method in protocol test[J]. Journal of Military Communications Technology, 2008, 29(1): 50-52. (in Chinese)

- [9] 谢磊, 魏蛟龙, 朱光喜. 基于改进 FSM 的协议一致性测试方法[J]. 通信学报, 2011, 32(6): 172-176.
XIE L, WEI J L, ZHU G X. Improved FSM-based method for protocol conformance testing[J]. Journal on Communications, 2011, 32(6):172-176. (in Chinese)
- [10] 马秀飞. 协议一致性测试与测试序列生成算法研究 [D]. 南京:南京邮电大学研究生部. 2005.
MA X F. Research on protocol conformance test and test sequence generation algorithm [D]. Nanjing: Graduate faculty of Nanjing university of posts and telecommunications. 2005. (in Chinese)
- [11] Lombardi F, Shen Y N. Evaluation and improvement of fault coverage of conformance testing by UIO sequences [J]. Communications, IEEE Transactions on, 1992, 40(8): 1288-1293.

作者简介



陈守宁 男, 1987 年生, 江苏徐州人, 南京邮电大学博士研究生在读, 主要研究方向为多媒体信号处理与无线通信中信号处理。

E-mail: njuptchensn@gmail.com



郑宝玉 男, 1945 年生, 福建闽侯人, 南京邮电大学教授、博士生导师, 主要研究方向为智能信号处理、通信信号处理和量子信号处理等。

E-mail: zby@njupt.edu.cn

李璟 女, 1987 年生, 江苏扬州人, 2013 年毕业于南京邮电大学通信工程专业, 获硕士学位。现供职于江苏移动南京分公司。E-mail: 1010010402@njupt.edu.cn

赵玉娟 女, 1979 年生, 山东青岛人, 2005 年毕业于西安电子科技大学, 获理学硕士学位。现为江苏教育学院数信院讲师, 南京邮电大学在职博士研究生, 目前研究方向为现代通信中的智能信号与信息处理。

E-mail: D0801@njupt.edu.cn