

移动 IPv6 测试中的层次化协议描述和测试生成方法

张玉军, 李忠诚

(中国科学院计算技术研究所, 北京 100080)

摘 要: 移动 IPv6 协议是一种庞大的分布式协议, 这为协议描述和测试生成带来了困难. 提出分层次描述协议的思想, 定义了描述协议的有限状态机和多节点有限状态机模型, 分四个层次分别对协议运行流程、协议节点类型、内部数据结构处理、离散功能进行了形式化描述. 把描述协议的状态机模型转化为有向图, 设计了针对有向图的测试序列集产生算法, 最终生成了覆盖整个移动 IPv6 协议的测试序列集.

关键词: 移动 IPv6; 一致性测试; 协议描述; 测试生成

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2004) 12A-030-05

Hierarchy Protocol Description and Test Generation Method for Mobile IPv6 Testing

ZHANG Yu-jun, LI Zhong-cheng

(Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China)

Abstract: Mobile IPv6 is complicated and distributed, which makes it difficult to describe protocol and generate test sequences. The idea of hierarchy protocol description is proposed. Finite state machine (FSM) and multi-node finite state machine (MN-FSM) are defined. Mobile IPv6 protocol is divided into four layers. FSM and MN-FSM are used to describe protocol performing procedure, mobile IPv6 nodes, internal data structure management and discrete functions. The state goto graphs are changed into directed graph. Test generation algorithm for directed graph is designed and the set of test sequences covering mobile IPv6 is generated.

Key words: mobile IPv6; conformance testing; protocol description; test generation

1 引言

移动 IP 提供了一种路由机制, 使得移动节点可以以一个永久的 IP 地址连接到任何链路上^[1-2]. 移动 IPv6 目前正处在不断发展的阶段, 新的技术不断被提出, 新的实现不断被开发出来. 随着支持移动 IPv6 的网络产品日益增加, 各实现与协议标准不一致性的问题也日益严峻, 对移动 IPv6 实现进行一致性测试的需求日益强烈.

一致性测试对保证协议实现与协议标准相一致, 并进而保证不同的协议实现之间能够互操作起着至关重要的作用^[3-4]. 测试生成是协议测试中的关键问题, 形式化技术是测试生成的基础^[5]. 移动 IPv6 协议是一种庞大的分布式协议, 这为协议描述和测试生成带来了困难. 目前阶段下, 大多数的测试研究组织都还没有开展移动 IPv6 测试研究和实践; 少数几个已经开展移动 IPv6 测试的组织也没有公布关于测试生成方面的研究成果, 大多采用手工生成测试例的方式, 测试集覆盖度得不到保证. 本文提出分层次对移动 IPv6 协议进行描述的思想, 对协议功能合理分层和解析并分别进行描述, 在此基础上实现测试序列自动产生, 解决了移动 IPv6 协议测试生

成问题.

2 层次化协议描述和测试生成

移动 IPv6 网络系统是一个复杂的分布式系统. 对分布式系统进行分层可以简化描述复杂度, 本文提出层次化协议描述和测试生成方法, 把整个移动 IPv6 协议分为四个层次加以描述并进行测试生成(图 1):

(1) 针对协议运行流程的描述和测试生成: 把整个移动 IPv6 网络看作一个封闭系统, 描述协议运行的所有流程并生成测试序列;

(2) 针对协议节点类型的描述和测试生成: 从被测主体的角度, 分别针对移动节点、家乡代理、通信节点、普通路由器

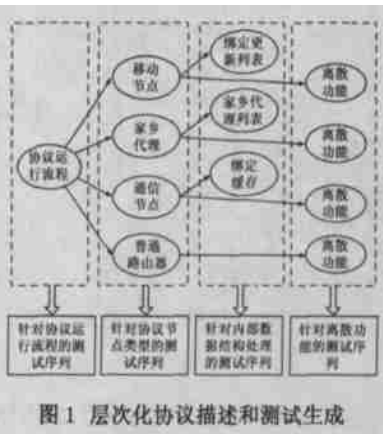


图 1 层次化协议描述和测试生成

的生命周期进行形式化描述和测试生成;

(3) 针对内部数据结构处理的描述和测试生成:主要考察协议实现在容错性等方面是否与协议标准一致;

(4) 针对离散功能的分析和测试:从提高测试覆盖的角度分析协议标准中定义的离散功能.

有限状态机 (Finite State Machine, FSM) 是最普遍的用于描述协议的形式化模型^[6]. 除标准有限状态机定义中必须包括的基本要素(状态集、输入、输出、变迁函数、输出函数)外, 为描述移动 IPv6 协议, 还需要定义行为函数, 描述系统在特定输入条件下发生的非输出、非状态转移的行为(表 1). 除有限状态机外, 为描述内部数据结构处理过程, 还定义了多节点有限状态机模型 (Multi-node Finite State Machine, MN-FSM). 状态机模型可以转化为有向图, 本文设计了针对有向图的测试序列集生成算法, 生成了针对整个协议的测试序列集.

表 1 有限状态机模型

有限状态机是一个六元组 $M = (S, I, O, f, g, h)$:	
1	$S = \{S_0, \dots, S_n\}$ 是有限状态集合;
2	$I = \{I_1, \dots, I_a\}$ 是有限输入事件集合;
3	$O = \{O_1, \dots, O_b\}$ 是有限输出事件集合;
4	$g: S \times I \rightarrow O$ 是该有限状态机的输出函数;
5	$f: S \times I \rightarrow S$ 是该有限状态机的状态转移函数;
6	$h: S \times I \rightarrow S$ 是该有限状态机的行为函数.

3 针对协议运行流程的描述

3.1 协议运行环境

一个典型的、最小化的移动 IPv6 网络系统至少应该包括以下 3 部分(图 2):

- (1) 链路类型: 应该存在三种类型的链路, 分别为家乡链路、外地链路、通信链路;
- (2) 路由器类型: 应该存在两种类型的

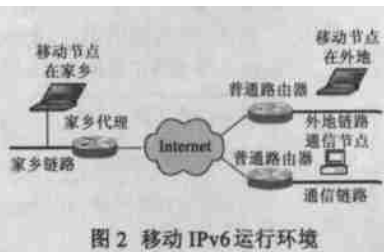


图 2 移动 IPv6 运行环境

路由器, 即普通路由器和家乡代理;

(3) 节点类型: 运行环境中应该有一个移动节点和一个或多个通信节点.

3.2 协议运行流程描述

基于图 2 描述移动 IPv6 运行环境, 可以从以下 3 方面对协议运行流程的状态进行描述:

- (1) 移动节点位置属性 $M_p = \{home, foreign\}$, home 表示移动节点在家乡, foreign 表示在外地;
- (2) 家乡注册成功标志 $HRflag = \{0, 1\}$, 1 表示移动节点在家乡代理上存在有效注册信息, 0 表示不存在;
- (3) 一般注册完成标志 $CRflag = \{0, 1\}$, 1 表示移动节点在一个或多个通信节点上存在有效注册信息, 且不存在未完成的注册, 否则为 0.

协议运行流程的当前状态可以用 $M_p, HRflag, CRflag$ 这三个属性进行描述, 设 S 为协议运行流程的状态集, 即 $S = M_p \times HRflag \times CRflag$. 集合 S 中共有 8 个元素, 去掉无效状态和不稳

定状态, 最终 $S = \{S_0, S_1, S_2\}$ (表 2).

表 2 协议运行流程状态说明表

S_0 (home, 0, 0)	初始状态, 移动节点在家乡链路上
S_1 (foreign, 1, 0)	移动节点在外地链路, 家乡注册完成、一般注册未完成
S_2 (foreign, 1, 1)	移动节点在外链路, 家乡注册和一般注册均已完成

表 3 针对协议运行流程的输入事件说明表

I_1	把移动节点从家乡链路移动到外地链路上
I_2	把移动节点从外地链路移动到家乡链路上
I_3	启用一个通信节点与移动节点通信

表 4 协议运行流程行为说明表

F_1	移动节点进行家乡注册
F_2	移动节点进行一次一般注册
F_3	移动节点解除家乡注册
F_4	移动节点解除所有当前有效的一般注册

移动 IPv6 网络是一个封闭系统. 对这个系统而言, 所有的输入都是人工事件(表 3), 不存在输出事件, 但是存在相应的行为(表 4). 图 3 描述了协议运行流程的状态转换过程.

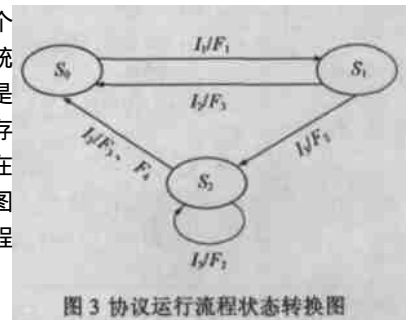


图 3 协议运行流程状态转换图

4 针对协议节点类型的描述

组成移动 IPv6 网络系统的基本元素包括移动节点、家乡代理、通信节点、普通路由器, 这些节点是真正被测试的主体. 本节仅以移动节点为例给出描述各种类型节点的方法, 着眼于各种类型节点在一个生命周期中针对不同外部输入条件的反应.

在实际运行过程中, 移动节点的所有行为与两个方面有关: 在家乡代理上维护注册信息和在通信节点维护注册信息. 在一个生命周期内, 移动节点的运行流程包括:

- (1) 正常情况下, 连接在家乡链路上, 处于稳定状态;
- (2) 检测到位置发生了移动(从家乡移动到外地), 生成转发地址;
- (3) 转发地址配置成功后, 执行家乡注册过程, 此后进入稳定状态;
- (4) 收到家乡代理通过隧道方式转发过来的数据包, 执行一般注册, 此后进入稳定状态;
- (5) 如果有其他通信节点与之通信, 重复一般注册过程;
- (6) 检测到位置发生了移动(从外地返回家乡), 解除家乡

注册和所有一般注册.此时一个移动周期完成.

针对移动节点描述,表 5 为状态说明,表 6 为输入事件,表 7 为输出事件,图 4 为状态转换图.

表 5 移动节点状态说明表

S_0	在家乡,没有注册信息
S_1	在外地,通过地址自动配置生成了转交地址,正在进行重复地址检测
S_2	向家乡链路发送了动态家乡代理地址发现请求消息,正在等待应答消息
S_3	在外地,向家乡代理发送了绑定更新消息,正在等待确认消息
S_4	在外地,且完成了家乡注册
S_5	在外地,完成了针对一个通信节点的一般注册过程
S_6	存在一般注册的情况下,回到家乡,正在解除家乡注册
S_7	不存在一般注册的情况下,回到家乡,正在解除家乡注册
S_8	出错状态

表 6 针对移动节点的输入事件说明表

I_1	移动节点第一次收到非家乡前缀信息
I_2	移动节点非第一次收到非家乡前缀信息
I_3	移动节点在外地,收到了关于家乡注册的绑定确认消息
I_4	移动节点回到家乡,收到了解除家乡注册的绑定确认消息
I_5	移动节点执行转交地址重复地址检测时,收到了其他节点返回的邻居宣告消息
I_6	移动节点收到了动态家乡代理发现应答消息
I_7	移动节点收到来自家乡代理的隧道包
I_8	移动节点收到家乡前缀信息
I_9	收到超时信息

表 7 移动节点输出事件说明表

O_1	发出用于家乡注册的绑定更新
O_2	发出隧道应答包和用于一般注册的绑定更新
O_3	回到家乡,发出用于解除家乡注册的绑定更新消息
O_4	回到家乡,发出用于解除一般注册的绑定更新消息
O_5	生成转交地址,发出用于重复地址检测的邻居请求消息
O_6	发出动态家乡代理地址发现请求消息
O_7	无输出

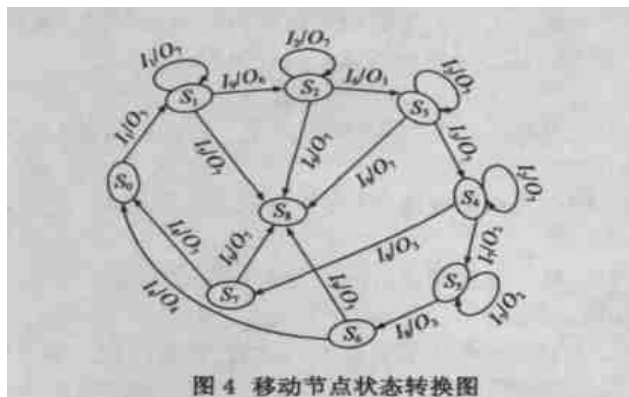


图 4 移动节点状态转换图

5 针对内部数据结构处理的描述

移动 IPv6 标准中定义了多种内部数据结构,包括绑定缓存、家乡代理列表和绑定更新列表,用于记录与移动相关的信息.对内部数据结构处理的测试主要是考察协议实现对各种输入条件(包括有效输入和无效输入)的响应,这是一种细力度的测试,也是测试的重点所在.限于篇幅,本节以绑定缓存处理为例说明如何对内部数据结构处理过程进行形式化描述.绑定缓存是家乡代理维护的一个重要内部数据结构,更是通信节点维护的唯一一个与移动相关的内部数据结构.绑定缓存由一个或多个条目组成,本节先对通信节点中单条绑定缓存条目处理进行形式化描述,在此基础上再对整个绑定缓存处理进行描述.

5.1 单条绑定缓存条目处理的描述

与一条绑定缓存条目有关的行为受到该条目有效生存时间控制,其在整个生存期内可能经历如下过程:

- (1)任何时刻收到绑定更新,都会重新设置有效生存时间(若无绑定缓存条目存在,则先创建条目);
- (2)有效生存时间到期后,通信节点发出绑定刷新请求;
- (3)通信节点发送数据之前,先查找绑定缓存条目,根据查找结果选择数据发送的方式;
- (4)收到删除绑定缓存条目的绑定更新、或者发出绑定刷新请求后没有收到应答,则删除绑定缓存条目;
- (5)任何时刻收到无效的绑定更新消息,缓存条目的状态不做改变.

表 8 描述了与单条绑定缓存条目处理有关的状态集,表 9 为输入事件,表 10 为输出事件,图 5 为状态转换图.

表 8 与单条绑定缓存条目处理有关的状态集 S_e

S_{e0}	绑定缓存条目不存在
S_{e1}	绑定缓存条目处在有效期内
S_{e2}	缓存条目即将过期,发出绑定刷新请求,正在等待绑定更新.

表 9 与单条绑定缓存条目处理有关的输入事件集 I_e

I_{e1}	收到有效的、生存期不为 0 的绑定更新
I_{e2}	收到用于删除绑定缓存条目的绑定更新
I_{e3}	针对绑定缓存条目有效时间的超时事件
I_{e4}	针对绑定刷新请求的超时事件
I_{e5}	收到无效的绑定更新

表 10 与单条绑定缓存条目处理有关的输出事件集 O_e

O_{e1}	针对收到的绑定更新,发出有效确认信息
O_{e2}	发出有效的绑定刷新请求

针对每一个缓存条目,设定一个变量 $V_{lifetime}$ 表示该条目最近更新过的有效时间;定义行为函数 f ,其功能在于根据收到的绑定更新消息重置变量 $V_{lifetime}$.绑定缓存条目处理中的状态变迁采用如下形式表示:

S(当前状态) I/F(行为)/O S(下一状态)

5.2 绑定缓存处理的形式化描述

在一个确定的运行/测试环境中,对于通信节点而言,可能与之通信的移动节点数量是确定的,因此绑定缓存条目的最大数量是确定的.提出一种多节点有限状态机模型 MN-FSM,用于对确定环境下通信节点中绑定缓存处理进行形式化描述(表 11).MN-FSM 是在 5.1 节中单条绑定缓存处理描述的基础上进行定义的.

表 11 多节点有限状态机模型

MN-FSM 是一个 10 元组 $S, I, O, N, q, f, V, bf, sf, S_0$, 其中:	
(1)	N 是该配置环境中移动节点索引的集合,如果存在 n 个移动节点,则 $N = \{1, 2, \dots, n\}$;
(2)	S 是绑定缓存的状态集,由 N 和 S_e 共同定义,每一个状态都是一个包含 n 个 N, S_e 元素的集合. $S_0 = \{1, S_{e0}, \dots, n, S_{e0}\}$;
(3)	I 是该环境下的输入事件集. $I \subseteq N \times I_e$;
(4)	O 是该环境下的输出事件集. $O \subseteq N \times O_e$;
(5)	q 是该环境下的输出函数. $q: S \times I \rightarrow O$;
(6)	f 是该环境下的状态转换函数. $f: S \times I \rightarrow S$;
(7)	V 是一个 $ N $ 元向量,每一个元素存储对应的绑定缓存条目最近设置的有效时间;
(8)	bf 是该环境下的行为函数,其作用是根据收到的有效绑定更新消息重置向量 V ;
(9)	sf 是该环境下的信号产生函数,其作用是产生 $N, \{I_{e3}, I_{e4}\}$ 输入事件.

下面以存在两个移动节点的运行环境为例,用 MN-FSM 对通信节点中绑定缓存处理进行描述.为简化说明,首先对图 5 中针对单条绑定缓存条目处理的描述进行简化,去掉无效输入 I_{e5} .则 $M = S, I, O, N, q, f, V, bf, sf, S_0$, 状态转换图见图 6:

(1) $N = \{1, 2\}$;

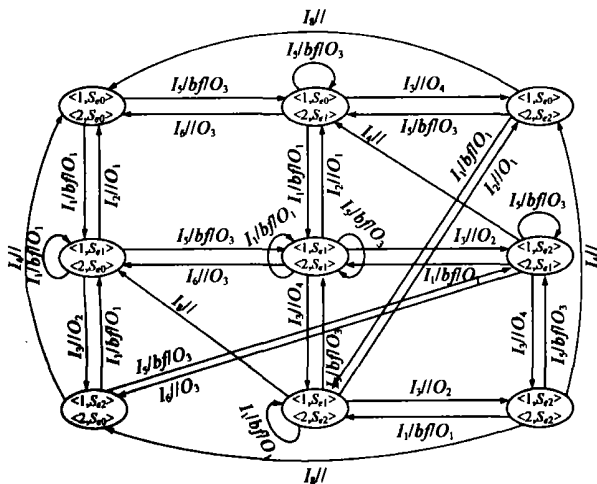


图 6 包含两个条目的绑定缓存处理的 MN-FSM

(2) $S = \{ \{1, S_{e0}, 2, S_{e0}\}, \{1, S_{e0}, 2, S_{e1}\}, \{1, S_{e0}, 2, S_{e2}\}, \{1, S_{e1}, 2, S_{e0}\}, \{1, S_{e1}, 2, S_{e1}\}, \{1, S_{e1}, 2, S_{e2}\}, \{1, S_{e2}, 2, S_{e0}\}, \{1, S_{e2}, 2, S_{e1}\}, \{1, S_{e2}, 2, S_{e2}\} \}$;

(3) $I = \{ I_1 = 1, I_{e1}, I_2 = 1, I_{e2}, I_3 = 1, I_{e3}, I_4 = 1, I_{e4}, I_5 = 2, I_{e1}, I_6 = 2, I_{e2}, I_7 = 2, I_{e3}, I_8 = 2, I_{e4} \}$;

(4) $O = \{ O_1 = 1, O_{e1}, O_2 = 1, O_{e2}, O_3 = 2, O_{e1}, O_4 = 2, O_{e2} \}$;

(5) $V = \{ 1, V_{lifetime}, 2, V_{lifetime} \}$;

(6) 状态转换函数、输出函数的定义遵循如下规则:如果在单条绑定缓存条目处理的状态机中有

$$S_{ei} \quad I_{ea} / f / O_{eb} \quad S_{ej}$$

那么在绑定缓存处理的多节点有限状态机中有

$$\{ 1, S_{ei}, 2, * \} \quad 1, I_{ea} / bf / 1, O_{eb}$$

$$\{ 1, S_{ej}, 2, * \}$$

和

$$\{ 1, *, 2, S_{ei} \} \quad 2, I_{ea} / bf / 2, O_{eb}$$

$$\{ 1, *, 2, S_{ej} \}$$

成立

6 针对离散功能的分析和测试

移动 IPv6 标准中定义了大量不能够进行形式化描述的离散功能,本节从测试的角度分析了这些离散功能,重点在于这些功能的应用和运行过程.明确了离散功能的应用,测试序列的构造就成为一项比较简单的工作了.移动 IPv6 协议中定义的离散功能包括:

(1) 为透明性考虑而定义的功能:在移动 IPv6 协议中,为了使移动功能对路由器和上层应用程序透明,定义了家乡地址目的选项和类型为 2 的路由报头;

(2) 为安全性考虑而定义的功能:IPSec 是网络层安全标准,提供网络层端到端通信的安全保护^[7].为了保证安全性,移动 IPv6 协议中许多地方都推荐或者强制采用 IPSec 对通信过程进行保护,包括家乡注册、返回可路由过程、移动前缀发现过程等;

(3) 为保证移动过程高效、可靠运行而定义的功能:为保证移动过程的可靠运行,协议标准中定义了绑定确认状态、绑定刷新建议选项、绑定缓存替换策略、绑定更新重传频率限制等机制.

7 测试生成算法和测试序列生成

为了更好地对描述协议的状态机进行机器处理,可以省略其中的输入/输出对,把状态机转化为有向图,图 7 为由移动节点状态机转化来的有向图,具有以下 3 特征:

(1) 存在一个节点,从其可以到达任何节点,该节点对应有有限状态机的初始状态,称为初始状态节点;

(2) 在某些节点上,存在自循环闭环.在状态机中,该节点对应的状态在特定输入条件下,不发生状态转换;

(3) 在设定了最大自循环次数的基础上,从任何节点开始

的有向路径最终会到达初始状态节点或者终端节点(在有向图中,出度为零的节点称为终端节点)。

在从有限状态机转化而来的有向图中,如下定义一条测试序列:

测试序列定义:在设定最大自循环次数的基础上,起始于初始状态节点,终止于初始状态节点或者终端节点的一条有向路径称为一条测试序列。所有测试序列构成了针对有向图的测试序列集。

测试生成的目的是得到有向图的测试序列集,采用多叉链表的形式存储有向图,表 12 为针对有向图的测试序列集生成算法。

表 12 针对有向图的测试序列集生成算法

<p>(1) 定义存储有向图节点的数据结构:</p> <pre> struct node { 该节点的标识信息 Identifier; 该节点后续节点的数量 Nextnode. Number; 指向该节点所有后续节点的指针列表 struct node * list[max]; } </pre> <p>(2) 采用多叉链表的形式存储有向图,对于每一个节点,顺序标注其所有的后续节点,把所有的后续节点编号为 1、2、……;</p> <p>(3) 定义一个堆栈 Stack,堆栈元素的格式为 节点,序号,用于存储节点和它的已经处理过的后续节点的序号,堆栈初始化为 $(S_0, 0)$;</p> <p>(4) 如果堆栈不空:</p> <p>(a) $(S_{v1}, i) = POP(stack)$;</p> <p>(b) 如果 $i + 1$ 不大于 S_{v1} 存在的后续节点的数量, $S_{v2} = nextnode(S_{v1}, i + 1)$;</p> <p>i. 如果 S_{v2} 为终端节点或者为初始节点 S_0,则堆栈中所有节点加上 S_{v2} 构成有向图的一条测试序列, $PUSH(S_{v1}, i + 2)$, 转 3;</p> <p>ii. 否则,如果 $S_{v1} = S_{v2}$, $PUSH(S_{v1}, i + 2)$, 转 3;</p> <p>iii. 否则, $PUSH((S_{v1}, i + 1), PUSH(S_{v2}, 0))$, 转 3.</p> <p>(c) 如果 $i + 1$ 大于 S_{v1} 存在的后续节点的数量,转 3.</p> <p>(5) 按照预先设定的最大自循环次数,对所有测试序列中的自循环节点进行预定次数的重复,得到最终的测试序列集;</p> <p>(6) 输出测试序列集。</p>
--

把测试序列集生成算法应用到图 7,设定最大子循环次数为 1,可得到如下针对移动节点的测试序列集:

- (1) $TS1 = S_0 I_1 O_5 S_1 I_2 O_7 S_1 I_5 O_7 S_8$;
- (2) $TS2 = S_0 I_1 O_5 S_1 I_2 O_7 S_1 I_9 O_6 S_2 I_2 O_7 S_2 I_9 O_7 S_8$;
- (3) $TS3 = S_0 I_1 O_5 S_1 I_2 O_7 S_1 I_9 O_6 S_2 I_2 O_7 S_2 I_6 O_1 S_3 I_2 O_7 S_3 I_9 O_7 S_8$;
- (4) $TS4 = S_0 I_1 O_5 S_1 I_2 O_7 S_1 I_9 O_6 S_2 I_2 O_7 S_2 I_6 O_1 S_3 I_2 O_7 S_3 I_3 O_7 S_4 I_2 O_7 S_4 I_8 O_3 S_7 I_9 O_7 S_8$;

$$(5) TS5 = S_0 I_1 O_5 S_1 I_2 O_7 S_1 I_9 O_6 S_2 I_2 O_7 S_2 I_6 O_1 S_3 I_2 O_7 S_3 I_3 O_7 S_4 I_2 O_7 S_4 I_8 O_3 S_7 I_4 O_7 S_0$$

$$(6) TS6 = S_0 I_1 O_5 S_1 I_2 O_7 S_1 I_9 O_6 S_2 I_2 O_7 S_2 I_6 O_1 S_3 I_2 O_7 S_3 I_3 O_7 S_4 I_2 O_7 S_4 I_7 O_2 S_5 I_7 O_2 S_5 I_8 O_3 S_6 I_9 O_7 S_8$$

$$(7) TS7 = S_0 I_1 O_5 S_1 I_2 O_7 S_1 I_9 O_6 S_2 I_2 O_7 S_2 I_6 O_1 S_3 I_2 O_7 S_3 I_3 O_7 S_4 I_2 O_7 S_4 I_7 O_2 S_5 I_7 O_2 S_5 I_8 O_3 S_6 I_4 O_4 S_0$$

8 结论

形式化技术是可验证测试生成的基础。通过把整个移动 IPv6 协议标准分成四个层次进行形式化描述并进行测试生成,得到了覆盖整个协议的可验证测试序列集,自动生成的测试序列集是设计移动 IPv6 测试集的基础。利用本文的研究成果,我们设计了完整的移动测试集,共包括 73 个测试例、102 个测试项目,覆盖了协议标准中 100% 的必备功能和绝大部分的可选功能。通过比较发现,TAHI 组织公布的手工生成的移动 IPv6 测试集只是我们生成测试集的一个真子集,这种比较结果验证了本文提出的层次化协议描述和测试生成方法的有效性。

参考文献:

- [1] 裴晓峰,等(译). 移动 IP[M]. 北京:机械工业出版社,2000.
- [2] D Johnson, et al. Mobility Support in IPv6[S]. IETF RFC 3775, 2004.
- [3] 王乐春,等. 高端路由器测试技术[M]. 北京:人民邮电出版社, 2002.
- [4] IT-OSF Conformance testing methodology and framework[S]. ISO/IEC 9646, 1996.
- [5] Ural H, et al. A uniform test derivation method for Estelle, LOTOS and SDL[A]. ISO/IEC JTC1/SC21/WGI meeting on Formal Methods in Conformance Testing[C]. 1993.
- [6] 古天龙,等. 网络协议的形式化分析与设计[M]. 北京:电子工业出版社, 2003.
- [7] S Kent, et al. Security Architecture for the Internet Protocol[S]. IETF RFC 2401, 1998.
- [8] 孙利民,等. 移动 IP 技术[M]. 北京:电子工业出版社, 2003.

作者简介:



张玉军 男,1976 年出生于河北省衡水,2004 年在中国科学院计算技术研究所获得博士学位,研究方向为下一代互联网协议、协议测试。
E-mail: zhmj @ict. ac. cn.

李忠诚 男,1962 年生于哈尔滨市,中国科学院计算技术研究所研究员,博士生导师,主要研究方向为计算机网络、容错计算。