IPv6 邻居发现协议的形式化验证*

叶新铭[1+], 郝松侠[2]

[1](内蒙古大学 计算机学院,内蒙古 呼和浩特 010021)

[2](中国科学院 软件中心,北京 100080)

Formal Verification of IPv6 Neighbor Discovery Protocol

YE Xin-Ming[1+], HAO Song-Xia[2]

[1](Collage of Computer Science, Inner Mongolia University, Huhhot 010021, China)

[2](Software Center, The Chinese Academy of Sciences, Beijing 100080, China)

+ Corresponding author: Phn: +86-471-4992931, E-mail: xmy@imu.edu.cn, http://imu.edu.cn

Received 2003-09-28; Accepted 2004-05-08

Ye XM, Hao SX. Formal verification of IPv6 neighbor discovery protocol. *Journal of Software*, 2005,16(6): 1182−1189. DOI: 10.1360/jos161182

Abstract: This paper presents the formal verification of properties of neighbor discovery protocol of IPv6 protocol suite using model checking. The protocol is modeled in MSC, whose use is popular in designing and documenting communication protocols. Linear temporal logic is adopted to specify properties of the protocol. The main result of this paper is an automatic method to extract properties from the MSC linearization directly.

Key words: message sequence charts; model checking; formal verification; automata theory

摘 要: 采用模型检查技术,对 IPv6 的邻居发现协议的属性进行了形式化验证.该协议的模型由目前广泛用于设计和描述通信协议的 MSC(message sequence charts)来描述,并通过线性时序逻辑说明该协议的属性.还提出了由 MSC 模型的线性化自动抽取协议属性的方法.

关键词: 消息序列表;模型检查;形式化验证;自动机理论

中图法分类号: TP393 **文献标识码**: A

随着网络和通信技术的高速发展,大量新的协议不断被提出.但是,目前的 Internet 的协议标准 RFC 仍使用自然语言描述.而且,随着网络服务要求的提高,使网络系统的复杂性在协议方面体现出空间分布性、并发性、异步性、不稳定性和多样性,网络协议再也不可能用工程直觉方法设计出高质量的协议,协议的完整性、正确性、安全性、可移植性和标准化都难以得到保证,协议实现后纠正协议描述错误的代价也是十分可观的.错误或不完全的协议可能导致整个网络的瘫痪,以及服务可靠性的降低.在这种情况下,迫切需要合适的方法、技术和计算机辅助工具来设计和维护网络协议.

* Supported by the National Natural Science Foundation of China under Grant No.60263002 (国家自然科学基金); the Key Science-Technology Project of Inner Mongolia under Grant No.2002061002 (内蒙古科技攻关项目)

作者简介: 叶新铭(1943−),男,内蒙古海拉尔人,教授,博士生导师,主要研究领域为网络协议形式验证与测试;郝松侠(1978−),女,硕士生,主要研究领域为网络协议形式验证与测试.

© 中国科学院软件研究所 http://www.jos.org.cn

长期以来,计算机协议都是验证的重点.协议设计常常引入细小的错误,除了少数协议运行以外,大部分这类错误都会被隐藏起来,这样将会导致更严重的操作错误.通过协议验证,可以保证协议设计的正确性和完整性,它主要验证协议的形式说明是否有逻辑错误.

传统的模拟方法虽然提供了丰富的资料,但不能提供完整的验证,而且费时较长.因此在协议验证方面,主要采用形式化验证技术模型检查和定理证明.模型检查是通过遍历系统的状态空间,验证系统模型是否具有相关的属性;定理证明是基于某个形式系统的公理推导出系统所具有的性质.

协议验证最早出现于 20 世纪 80 年代.IBM,Nortel,Intel,Motorola 等,都成立了自己的验证小组.2001 年,施乐公司的 PARC 研究中心、Agilent 实验室、IBM 华生实验室、微软研究院和斯坦福研究所,将协议形式化验证等技术作为近五年研究的重点.俄罗斯、日本等国家也分别投入资金开展了协议验证方面的工作.同时,还出现了一些国际化的组织,如官方的 IETF,ITU-U 及成立于 1960 年非官方的 IFIP,来推动验证技术的理论发展.一年一度的国际会议 VERIFY 以及 FORTE 等,也极大地促进了验证技术在计算机各个领域的应用.

我国的协议验证工作虽然起步较晚,但在短短十几年内,也取得了长足的发展.高技术研究发展计划以及国家自然科学基金会组织实施的网络与信息安全重大研究计划,均将协议验证作为重点研究项目.中国科学院计算技术研究所等科研院所和高等学校也开展了这方面的理论研究工作.

尽管如此,目前国内外的协议验证工作仍集中于如安全协议等有限的协议类型方面,且现有的验证工具也多是对协议实现进行验证,而非对协议设计本身的验证,致使当前的协议形式化验证技术存在一定的局限性.本文通过对 IPv6 邻居发现协议进行形式化验证,对协议的形式模型自动验证技术进行了研究.本文第 1 节给出 MSC(message sequence chart)的模型检查方法.第 2 节对邻居发现协议进行形式化验证,并对验证结果进行分析.第 3 节给出结论.

1 MSC 的模型检查

用于建立形式化模型的方法虽然有很多,如 Petri 网、SDL、LOTOS 等,但 MSC 是 IUT-T 提供的一种较新的形式描述语言,它是一种标准的迹语言.与传统的形式描述语言相比,它着眼于并发进程之间传递的消息,通过描述消息交换来说明系统成分与环境之间的通信行为.它以直观、透明的方式反映通信行为,省略了传统的形式描述技术必须考虑的大多数诸如程序变量以及变量值这样的细节,特别适用于说明通信协议.所以本文采用 MSC 为邻居发现协议建立模型.

描述 MSC[1]操作语义的方法有很多,本文中,我们通过考虑在线性时间域内的半序(partial order)语义,而非交叉语义,获得更自然、更简单的语义.这种符号语义基于半序多重集(partial-order multi-sets,简称 pomsets)[2],pomsets 是一个良好建立的线性时间非交叉模型.

对于 MSC 的模型检查问题,我们可以将其形式化为形式化验证的自动机理论方法[3].协议说明由一个 MSC、MSC 图或者 HMSC 来描述,其中的每个事件用一个字符集Σ来标记.系统的语义是Σ上的一个串语言.属性由Σ上的一个自动机来描述,该自动机的语言包含不期望的行为,则模型检查问题转化为检查两个语言的交集是否为空.

1.1 基本概念

一个消息序列表(MSC)M 是一个六元组[3]:$M=\langle V,<,L,T,m,P\rangle$,其中,$V$ 是一个事件集合;$<:\subseteq V\times V$;P 是一个进程集合;$L:V\to P$ 是将每个事件与一个进程相关联的映射;$T:V\to\{s,r,l\}$是一个描述事件类型(分别为 send,receive,或 local)的映射;$m\subset V\times V$是发送和接收事件之间的一个关系.

属性从语义上可定义为一个状态的无限序列集合,它表示了满足该属性的所有执行.有许多使用不同逻辑的时序逻辑模型检查过程[4],虽然我们使用线性时序逻辑,但所讨论的大部分思想都可以用于其他的逻辑和模型.该逻辑使用经典的布尔连接词(¬,∨,…)和时序操作符◇(eventually,也记做 F),□(always,也记做 G),○(next,也记做 X),U(until)来构造命题表达式.

© 中国科学院软件研究所 http://www.jos.org.cn

1.2 MSC的半序语义(partial order)

并发系统的每个单独执行都可以用两种方式建模:交叉语义(interleaving semantics)和半序语义(partial order semantics)[3].在交叉语义中,系统是一个线性执行的集合,需要在这些执行上对时序逻辑作出解释,系统满足一个表达式φ当且仅当它的所有执行都满足φ.在半序语义(或迹语义)中,每个执行都是一个半序,其中,事件的排序关系反映了事件之间的因果依赖性(causal dependency).两种语义之间的关系是,一个系统的交叉执行是一个半序执行的线性化.

图1中,我们以邻居发现协议的地址自动配置模块中,主机主动发送路由器请求的行为为例,给出该MSC模型的发送和接收事件的半序关系.为了直观表现这种关系,我们对该模型做了必要的化简,仅考虑最简单的情况,并省略了部分条件判断语句.

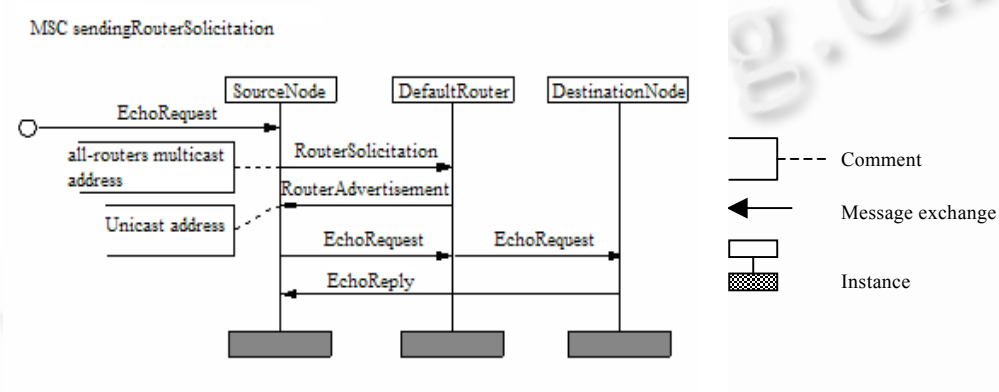

Fig.1 MSC of the host's active send behavior of router solicitation message

图1 主机主动发送路由器请求行为的MSC模型

该MSC模型描述了3个进程SourceNode,DefaultRouter, DestinationNode之间的交互活动.当SourceNode接收到外界的EchoRequest消息时,触发SourceNode主动发送多播路由器请求消息RouterSolicitation来定位默认路由器.DefaultRouter通过单播路由器宣告消息RouterAdvertisement给SourceNode,提供自己的路由信息,SourceNode将其确定为自己的默认路由器.SourceNode发送给DestinationNode的EchoRequest消息,由DefaultRouter进行转发,此时,DestinationNode发送EchoReply消息给SourceNode作为响应.

对于两个事件e和f,有$e<f$,当且仅当满足下列任意一条:

① e和f是同一个消息的发送和接收事件.在这种情况下,称e和f是一个消息对(message pair);

② e和f属于同一个进程p,且在进程线上e在f之前出现.这样,对于每个进程p,强加了它的所有事件的一个全序.由此我们可以得到如图2所示的半序图.

1.3 MSC的线性化(linearizations of MSCs)

对MSC的线性时序逻辑LTL属性的检查,是基于MSC的线性化的,即我们需要将事件间的半序(partial order)转为全序(total order),这样MSC执行的所有线性化都满足给定的LTL说明[5].每个线性化都是将一个MSC执行完全为一个全序.

我们将每个事件用一个序对来表示,该序对包括事件的类型(send或receive)及相应的进程.为了便于说明,将图1中的进程SourceNode,DefaultRouter,DestinationNode分别记为$P1$,$P2$,$P3$,该图的执行有唯一的线性化结果:(s,$P1$) (r,$P2$) (s,$P2$) (r,$P1$) (s,$P1$) (r,$P2$) (s,$P2$) (r,$P3$) (s,$P3$) (r,$P1$).

© 中国科学院软件研究所 http://www.jos.org.cn

每个线性化都有向前边(forward edges),该边从每个发送节点到与之匹配的接收节点.在上面的线性化例子中,由每个发送事件出发的向前边总是指向它在序列中的直接后继(immediate successor)[3].在一半序$\zeta=(V,<,\rightarrow)$的一个线性化ξ上,一个 LTL 表达式φ的解释遵循在ζ中φ的解释.也就是说,按照上述的半序语义,若ξ的第i个事件满足φ(记为$(\xi,i)|=\varphi$),则ζ的第υ个事件也满足φ(记为$(\zeta,\upsilon)|=\varphi$).

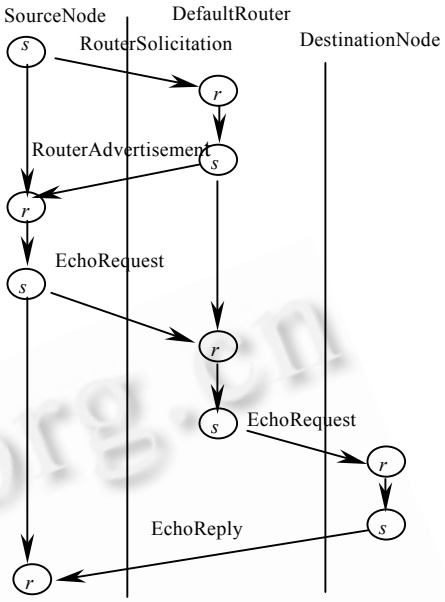

Fig.2 Partial order semantics of Fig.1

图2 图1的半序语义

1.4 自动机理论模型检查

Büchi 自动机是一个六元组$\langle S,S_0,\Sigma,L,\delta,F\rangle$,其中 S 是状态的有限集合,$S_0\subseteq S$ 是初始状态,Σ是有限字符集,$L:S\rightarrow\Sigma$是由状态到字符的映射,$\delta\subseteq S\times S$ 是转换关系,$F\subseteq S$ 是接受状态(accepting states).

属性自动机A是表示被禁止执行集合的 Büchi 自动机.说明自动机是描述协议说明的受限的 Büchi 自动机,它要求所有的状态都是接受状态.一个说明自动机和一个属性自动机的乘积(或交集)给出了说明中不满足该属性的所有执行.因此,只有协议说明不满足该属性时,二者的交集才不为空.此时,交集中的任何序列都可以用作反例.为了检查乘积 Büchi 自动机是非空的,我们需要检查在该自动机中是否存在一个包含一个接受状态,并且由初始状态可达的循环.并不需要检查乘积自动机的所有可能的循环,只要检查它是否包含至少一个由初始状态可达,并包含 F 中的一个状态的最大强连通成分.

1.5 MSC的模型检查

MSC M 说明了它所包含的事件的半序,M 的模型检查问题,可以通过构建适当的自动机,这个自动机接收这个半序的所有可能的线性化;检查这个自动机以及给定说明的属性自动机.这个问题是 coNP-complete 的[6].

对于Σ-标记 MSC M,语言$L(M)$表示系统所有可能的执行.我们可以用Σ上的自动机A来说明待验证的属性,这一自动机接收所有不希望的执行,则模型 M 满足属性 A,当且仅当交集 $L(M)\cap L(A)$为空.由此,我们可以得到 MSC 的模型检查问题:给定Σ-标记 MSC M,和Σ上的自动机A,确定二者语言的交集$L(M)\cap L(A)$是否为空.

为了解决模型检查问题,我们可以用从半序中抽取全局状态的标准技术,来构建自动机 A_M,A_M 识别给定 MSC 的每个执行的单独线性化,它可以接收$L(M)$.

由 MSC 构造自动机A_M的方法[6]是:割集(cut) c 是 E 关于<闭合的子集:若 $e\in c,e'<e$,则 $e'\in c$.因为一个单独进程的所有事件都是线性排序的,所以可以用元组来描述割集,这个元组给出了每个进程的最大事件(maximal event).自动机A_M的状态对应这些割集,空割集作为初始状态,包含所有事件的割集作为终态.如果割集d等于割集c加单独事件e,则由c到d有一条边,标记为$l(e)$.很容易验证自动机A_M接收语言$L(M)$.或者,我们也可以从一个执行的线性化出发,通过将A_M的一个执行的第i个节点标记为线性化中的第i个事件的命题来构建A_M.

综上所述,给定 MSC 模型M和时序逻辑表达式f,我们将其模型检查过程归纳为:

① 为M构造有限 Büchi 自动机,记为A_M;

② 为f的取反表达式$\neg f$的无限字构造有限 Büchi 自动机,得到的自动机记为$A_{\neg f}$;

③ 计算乘积自动机$A_G=A_M\times A_{\neg f}$(实际应用时,检查可达状态即可);

④ 检查自动机A_G是否为空.

至此,我们已经给出了将 MSC 转换为 Büchi 自动机,并采用自动机理论方法解决 MSC 的模型检查问题的方法.下一节将给出该模型检查方法在我们所作的邻居发现协议的形式化验证工作中的应用.

© 中国科学院软件研究所 http://www.jos.org.cn

2 邻居发现协议的形式化验证

邻居发现协议是 IPv6 协议的一个基本的组成部分,RFC2461[7]是发现协议的标准文本,它是由因特网工程任务组(IETF)在 1998 年 12 月制定的,它实现了在 IPv4 中的地址解析协议(ARP)、控制报文协议(ICMP)中的路由器发现部分、重定向协议的所有功能,并进行了一定的扩展.

2.1 协议的正确性属性(correctness properties)

并不存在完全“正确”的协议,也就是说,我们只能针对某个特性的正确性需求,称该协议是正确的.

在协议验证中,主要关心两类正确性属性:安全性(safety properties)和活性(liveness properties).非形式地,安全属性表示在系统运行期间,“坏的事情将不会发生”.活性表示在运行期间,“好的事情终将发生”.

另外,根据邻居发现协议的工作机制,即通过发送 ICMP 信息包来发现邻居节点,从而保证数据包的高速有效传输.我们发现,消息的传递和交换是保证邻居发现协议正确运行的根本.所以,在我们的验证工作中,还需要考虑这种消息间的响应关系.虽然它们仍然属于传统意义上的活属性,但是作为我们验证工作的重点,对其进行了重新定义,将其定义为“响应属性(response property)”,即节点对于请求作出响应的能力.

由于本文只关心主机的行为属性,表 1 列出了我们所选取的有关主机的所有属性.第 1 列为我们所划分的 3 个属性类别:响应属性、活属性和安全属性.第 2 列给出了属性的描述,其中对每个属性类别做了进一步的细化,对每个小的分类,给出了一个通用的逻辑表达式,例如,我们将响应属性类别分为两类逻辑表达式 $G(req\rightarrow F\ ack)$ 和 $G(rec\ A\rightarrow \exists\bigcirc send\ B)$.进行验证时,我们将每个通用逻辑表达式具体化.如第 2 列第 1 行的属性描述:正常情况下发送 *NS*,其对应的逻辑表达式为 $G(NS\rightarrow F\ NA)$,该属性的 3 种情况,我们可以使用带参的消息来描述.

关于安全属性和活属性的判断和识别方法,文献[8]已经做了大量研究,下面我们给出响应属性的选取方法.在给出算法之前,先进行两点说明.

首先,根据邻居发现协议的非形式描述,我们发现协议所使用的 ICMP 消息,可以根据其请求-响应关系构成匹配消息对.因此,我们定义匹配消息序对集合 $MP=\{(\text{EchoRequest},\text{EchoReply}),(NS,NA),(RS,RA)\}$.

另外,我们将第 1.3 节中的 MSC 线性化描述进行扩展.将原有的序对形式(e,p)扩展为三元组形式(e,p,m),其中 e 为事件类型 send 或 receive,p 为该事件对应的进程,m 为该事件对应的消息.

利用匹配消息对集合和 MSC 的线性化描述,我们得到响应属性的选取算法.由前述可知,属性是一个序列集合,记为 *ps*.对于 MSC 的每个线性化执行ξ,若其长度为 n,则定义属性序列 *ps* 的首元素为ξ的第 1 个三元组,记为(e_0,p_0,m_0),最终得到的属性序列放入属性集合 *PS*.对于序列中的三元组(e_i,p_i,m_i),$1<i\leq n$,有

1) 若 $e_i=\text{receive}$,$p_i=p_0$,则判断(m_0,m_i)是否属于匹配消息序对集合 *MP*;

2) 判断结果为:

a. 若属于 *MP*,则将(e_i,p_i,m_i)加入属性序列 *ps*,判断得到的 *ps* 是否已存在于属性集合 *PS*,若无,则将其作为一条待验证的响应属性,加入 *PS*;

b. 若不属于 *MP*,则 $i=i+1$,转 1);

3) $i=i+1$,若 $i<n$,则转 1);

4) 若 $i=n$,且 $ps=(e_0,p_0,m_0)$,则该线性化ξ无对应的响应属性;否则,*PS* 为ξ对应的响应属性集合.

由于图 1 仅存在一个线性化结果(见第 1.3 节),所以将上述算法应用于该线性化结果,即可得到图 1 的 MSC 模型所对应的响应属性.根据第 1.5 节的线性化结果,所抽取的属性序列 *ps* 的首元素为此线性化结果的初始元素$(s,P1,RS)$,且图 1 的线性化结果中,仅有$(s,P1,RS)$与$(r,P1,RA)$构成匹配关系.由此,我们可以得到该线性化结果的唯一属性 $G(RS\rightarrow F\ RA)$,对应的属性序列为$(s,P1,RS)$ $(r,P2,RS)$ $(s,P2,RA)$ $(r,P1,RA)$ $(s,P1,\text{EchoRequest})$ $(r,P2,\text{EchoRequest})$ $(s,P2,\text{EchoRequest})$ $(r,P3,\text{EchoReply})$ $(s,P3,\text{EchoReply})$ $(r,P1,\text{EchoReply})$.其中,为了确定验证结果,我们增加了确认序列$(s,P1,\text{EchoRequest})$ $(r,P2,\text{EchoRequest})$ $(s,P2,\text{EchoRequest})$ $(r,P3,\text{EchoReply})$ $(s,P3,\text{EchoReply})$ $(r,P1,\text{EchoReply})$.

© 中国科学院软件研究所 http://www.jos.org.cn

2.2 验证过程

我们以图 1 为例,利用第 1.3 节中的线性化结果和第 1.5 节中的 Büchi 自动机的构造算法,给出邻居发现协议的验证过程.

为了便于描述由MSC得到的自动机变迁,首先引入两个字符集$\Sigma_!$和$\Sigma_?$[9].其中,M是消息集合,P是进程集合.

(1) $\Sigma_! = \{!\} \times P \times M \times P$ 是发送动作集合,它的元素 $\langle !, p, m, q\rangle$ 表示为 $!_{p,q}\, m$.

Table 1 Property list of host

表 1 主机属性列表

Property category	Property description			
Response property	$G(req \rightarrow F\ ack)$: Every request message will be followed by an acknowledge message.	Send NS	Send NS (neither multicast nor unicast)	Local link address⇒Local link address
				Global link address⇒Global link address
				Global link address⇒Local link address
			Multicast NS	Default configuration.
			Unicast NS	Change the configuration of retransmission time.
		Send neighbor broadcast message NA		
		Send router solicitation message RS		
		Send router broadcast message RA		
		Send redirect message		Redirect to host
				Redirect to router
	$G(recA \rightarrow \exists \bigcirc sendB)$: Send message B immediately after receiving message A.	Receive NS	Receive valid neighbor solicitation message NS	
			No existing neighbor cache entry for this address	
			The state of the matching neighbor cache entry is:	INCOMPLETE
				REACHABLE
				STALE
				DELAY
				PROBE
			The IsRouter flag is set in the received NS	
		Receive RA	Receive non-solicited RA, and send router solicitation message RS	
			Receive non-solicited RA, and doesn't send router solicitation message RS	
		Receive redirect message		
Liveness property	G(addr-res$B$ U ncexist A):Performing address resolution for address B until creating the neighbor cache entry A			
	G(rec$A \rightarrow F$ nexthop B):Receive message A, and make the next-hop determination until finding the next hop B.			
Safety property	$G(\neg$(deadlock$\vee$livelock)):No live lock and dead lock exists.			
	$G(\neg$rec invalid A): Will not receive invalid message A	Receive invalid NS		
		Receive invalid NA		
		Receive invalid redirect message		
	$\neg F$(tran$A \wedge F$(tran $B \wedge F$ tran A)):Transaction B should not interfere with the events of transaction A.			
	G(begin$A \rightarrow \exists \bigcirc$(tran $A \cup$finish A)):The execution of transaction A will not be interfered by the other events.			

(2) $\Sigma_? = \{?\} \times P \times M \times P$ 是接收动作集合,它的元素 $\langle ?, p, m, q\rangle$ 表示为 $?_{p,q}\, m$.

集合 $\Sigma_c = \Sigma_? \cup \Sigma_!$ 定义为通信集合(communications set).MSC M 的事件 e 可以映射到 Σ_c ,标记为 l(e).

利用第 1.5 节中割集的概念,先构造图 1 的 MSC 模型的 Büchi 自动机形式 A_M.由第 1.2 节中半序关系<的定义可知,图 1 的事件集合 E 的半序关系为:

$!_{P1,P2}RS?_{P2,P1}RA<!_{P1,P2}Erq<?_{P3,P1}Erp,?_{P1,P2}RS<!_{P2,P1}RA<?_{P1,P2}Erq<!_{P3,P1}Erp,!_{P1,P2}RS<?_{P1,P2}RS,!_{P2,P1}RA<?_{P1,P2}RA,$
$!_{P1,P2}Erq<?_{P1,P2}Erq,!_{P2,P3}Erq<?_{P2,P3}Erq,!_{P3,P1}Erp4<?_{P3,P1}Erp.$

首先,第 1 个事件$!_{P1,P2}RS$作为割集 $C1$ 的初始元素,由于事件集合 E 中无更小事件,所以 $C1$={$!_{P1,P2}RS$}.其次,将事件$?_{P1,P2}RS$ 作为割集 $C2$ 的初始元素,根据上述半序关系,仅有事件$!_{P1,P2}RS$<事件$?_{P1,P2}RS$,则可得

© 中国科学院软件研究所 http://www.jos.org.cn

$C2=\{!_{P1,P2}RS,?_{P1,P2}RS\}$.依此类推,可得表 2 所示的割集,为了直观起见,我们用最大事件来描述每个割集.由此可得到图 1 的自动机 A_M (如图 3 所示),其状态所对应的割集见表 2.

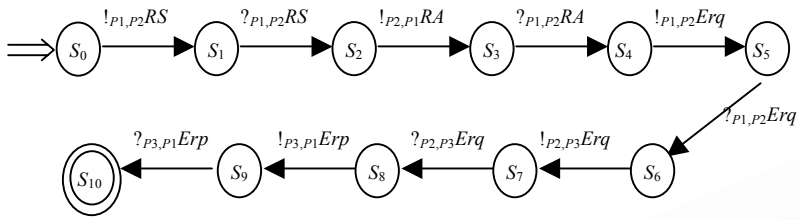

Fig.3 Büchi automata of Fig.1

图3 图1的 Büchi 自动机形式

Table 2 State description of Fig.3

表 2 图 3 的状态描述

State	Cut	Max. event of cut
S_0	$\varnothing$	Null
S_1	$C1$	$P1$ sends RS message to $P2$
S_2	$C2$	$P2$ receives RS message from $P1$
S_3	$C3$	$P2$ sends RA message to $P1$
S_4	$C4$	$P1$ receives RA message from $P2$
S_5	$C5$	$P1$ sends EchoRequest message to $P2$
S_6	$C6$	$P2$ receives EchoRequest message to $P1$
S_7	$C7$	$P2$ sends EchoRequest message to $P3$
S_8	$C8$	$P3$ receives EchoRequest message from $P2$
S_9	$C9$	$P3$ send s EchoReply message to $P1$
S_{10}	E	$P1$ receives EchoReply message from $P3$

其次,构造图 1 对应的取反属性的自动机.该属性的时序逻辑表达式为 $f=G(RS\rightarrow F\ RA)=\neg RS\vee(F\ RA)$,其取反表达式 $\neg f=RS\wedge\neg RA$.根据文献[10]给出的翻译算法,得到 $\neg f$ 的 Büchi 自动机形式 $A_{\neg f}$,如图 4 所示.

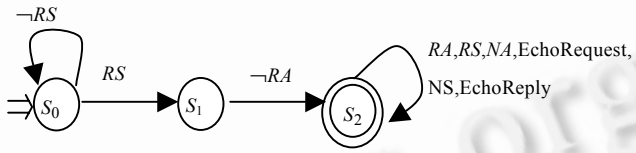

Fig.4 Büchi automata of negative property of $G(RS\rightarrow F\ RA)$

图 4 属性 $G(RS\rightarrow F\ RA)$取反表达式的 Büchi 自动机

最后,判断图 3 和图 4 的乘积自动机 $A_G=A_M\times A_{\neg f}$是否为空.由乘积自动机[3]定义可知,不存在一个包含 F'的接受状态子集,并且由初始状态可达的循环,故 A_G 为空,即图 1 的 MSC 模型满足属性 f.

依此类推,我们可以得到表 1 的所有属性的验证过程.

2.3 结果分析

由于本文主要考虑主机的行为,通过对第 1 建立的 MSC 模型进行模型检查,我们发现邻居发现协议说明中在描述主机行为时,存在一些细小的不足,但这并不会影响协议的整体运行.

协议说明中邻居缓存的 STALE 和 DELAY 状态的描述缺少足够的细节,使得我们无法确知,主机进入这两个状态后可能存在的行为,而且在某些情况下,无法对这两个状态加以严格的区分.

协议说明中对于定时器的描述,如 ReachableTime 等时间值,只是给出了概念性的定义,随着网络拓扑的改变,我们不可能使用同样的定义,这就可能导致某些未预料到的消息的接收和发送动作.

当然,我们对协议进行形式化验证的目的,主要是为了发现其中潜在的问题,从而在真正的实现中解决这些问题.所以,上述验证过程中存在的两个主要问题,我们都可以在具体实现时进一步地补充细节.比如,微软在实

© 中国科学院软件研究所 http://www.jos.org.cn

现其 Windows 2000 操作系统下的 IPv6 协议栈时,将邻居发现协议中邻居缓存的 STALE 和 DELAY 状态进行了合并.

3 结 论

本文给出了用模型检查对 IPv6 的邻居发现协议进行形式化验证的自动化方法,并对验证结果进行了一定的分析.在验证过程中,我们使用目前国际流行的形式说明语言 MSC,为邻居发现协议建立模型.为了减轻描述协议属性时可能产生的问题,我们采用了基于半序语义,而非传统的交叉语义的时序逻辑来说明待验证的属性.当然这只是一种可能的选择,我们还可以使用其他的逻辑来说明属性.

经过实验我们发现,虽然形式化验证增加了协议开发早期的开销,但是它完全可以发现协议设计中存在的潜在错误,从而在协议的实现过程中,通过必要的手段,避免或改正这些错误,降低了协议实现后可能出现的错误的迭代增长.

References:

[1] ITU-T.ITU-T Recommendation Z.120, Message Sequence Chart (MSC), 1999.

[2] Pratt V. Modeling concurrency with partial orders. Int'l Journal of Parallel Programming, 1986,15(1):33−71.

[3] Peled D. Specification and verification of message sequence charts. In: Bolognesi T, Latella D, eds. Proc. of the Formal Methods for Distributed System Development. Pisa: Kluwer Academic Publishers, 2000. 139−154.

[4] Alur R, Peled D, Penczek W. Model-Checking of causality properties. In: Proc. of the Proc. of the 10th Annual IEEE Symp. on Logic in Computer Science. San Diego: IEEE Computer Society Press, 1995. 90−100.

[5] Muscholl A, Peled D, Su Z. Deciding properties for message sequence charts. In: Nivat M, ed. Proc. of the FoSSaCS 1998, LNCS 1378. Berlin: Springer-Verlag, 1998. 226−242.

[6] Alur R, Yannakakis M. Model checking of message sequence charts. In: Baeten JCM, Mauw S, eds. Proc. of the CONCUR'99, Concurrency Theory. LNCS 1664, Berlin: Springer-Verlag, 1999. 114−129.

[7] Narten T, Nordmark E, Simpson W. Neighbor discovery for IP version 6 (IPv6). RFC 2461, 1998.

[8] Kindler E. Safety and liveness properties: A survey. EATCS-BuUetin, 1994,53:268−272.

[9] Deussen PH, Tobies S. Formal test purposes and the validity of test cases. In: Peled DA, Vardi MY, eds. Proc. of the FORTE 2002. LNCS 2529, Berlin: Springer-Verlag, 2002. 114−129.

[10] Holzmann HJ. The model checker spin. IEEE Trans. on Software Engineering, 1997,23(5):279−295.

© 中国科学院软件研究所 http://www.jos.org.cn