

网络协议中的时间约束测试

陈东洛, 尹 霞, 吴建平

(清华大学计算机系, 北京 100084)

摘 要: 对协议实现的时间约束进行测试, 是验证协议实现的实时特性的重要方法. 本文考查了当前实时系统测试方面的进展, 结合路由协议测试的实践, 修正了现有方法几个不符合实际情况的假设, 扩展了时间自动机的理论和思路, 将时间约束下的状态机转换成非确定性有限状态自动机, 运用 W_p 方法生成抽象测试集, 讨论将抽象测试集参数化的方法. 以路由协议 RIP 的时间约束为例说明了生成基于 TTCN 测试例的过程.

关键词: 时间自动机; 协议测试; RIP; 测试生成 W_p 方法; TTCN

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2002) 12A-1947-06

Testing Time Constraints in Network Protocols

CHEN Dong-luo, YIN Xia, WU Jian-ping

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

Abstract: Time constraints are common in network protocols and are important features for real-time requirements. It is necessary to test time constraints in protocol implementations. Approaches in testing real-time are reviewed and the disadvantages are given. Some assumptions in the existing methods are revised to fit the need of our practice on testing routing protocols. We modified the theory of timed automata to convert time constraints to state variables and time events, which result in a nondeterministic finite state machine. Abstract test suite is generated by W_p method and should be parameterized in execution. Example from RIP protocol is used to demonstrate the procedure to construct TTCN test cases.

Key words: timed automata; protocol testing; RIP; W_p method; TTCN

1 引言

网络协议的一致性测试是检验网络设备功能正确性的重要手段. 大多数网络协议在规定通讯实体之间交互规则的时候, 用时钟约束来表示操作的时间限制. 对网络协议的时间约束进行测试就是检验协议实现中的时间是否遵循规范. 在路由协议中, 路由进程为每条路由设置时钟, 来计算该路由的有效性. 在多媒体传输过程中, 视频信号、音频信号和文本信号是分别传送的, 它们之间需要满足实时和同步的要求^[1]. 这些具有实时性要求的协议中, 除了消息交互的顺序需要符合规定以外, 还要保证这些消息是在规定的时间范围发生. 随着多媒体传输和移动通讯的这些实时服务需求的增加, 对实时系统的测试需求也在增加.

目前, 国内外上对于实时系统验证已经开展了大量工作, 而实时系统测试方面的研究正在开展^[1-4]. 不少研究机构都将测试实时系统列入规划. 文[1]测试多媒体传输中不同流之间的时间约束和同步问题. 文[2,4]要解决一般实时系统测试问题, 两者使用相似的方法, 将被测系统转换成区域自动机, 通过对区域自动机采样, 提取出点格自动机(Grid Automata);

点格自动机实际上是一个有限状态机, 可以应用已有的测试生成算法来生成测试集. 但是这种方法中有一个缺点, 采样需要一个额外的假定, 即要求输出发生在整数时刻. 这个假设不符合网络协议实际的情形, 网络协议中为了避免突发流量, 往往允许输出等待一个小的随机时间.

现有的文献只处理了确定性情况下的测试生成, 文[2,3]中要求不能有两个输出时间, 或者一个输入事件和一个输出事件, 同时满足时间约束. 但是在路由协议中, 路由更新存在随机性, 协议本身允许不确定性的行为.

本文结合路由协议测试实践, 分析时间约束测试问题, 修正了已有工作中一些不合理的假定, 在现有时间自动机的方法基础上, 添加时间变迁, 将带有时间约束的有限状态机转换成非确定性状态机, 应用 W_p 方法实现了不确定性情况下的测试生成. 对时间变迁进行参数化, 直接生成 TTCN 形式的测试例. RIP 协议路由更新中的时间约束行为作为举例贯穿全文.

2 时间输入输出自动机的基本概念

时间模型可以分为离散时间和连续时间. 离散时间模型

适合于能够控制节拍的系统,例如在数字电路,能够直接用有限状态机来描述.在计算机网络中,协议的行为不是按照节拍来发生的,离散时间模型不适用于网络协议中的实时行为描述.连续时间(dense-time)模型能够表达此类系统时钟的连续变化,系统内部包含一组时钟,并按照同样的速率增长.系统的行为往往带有时间约束,时钟取不同值时系统的行为是不一样的,时钟实际上是系统状态的一部分.在理论上时钟是连续的,具有无穷多个值,系统也就有无穷多的状态.解决这个问题的思路是将时间空间划分成有限个等价类,每个等价类(成为区域)中系统具有相同的时间.系统的状态由这些等价类构成,这就成为区域自动机(Region Automata).

常用的描述时间系统的模型有时间 Petri 网(TPN)^[5],时间变迁系统^[6],时间自动机等.TPN 和时间变迁系统的主要原则是在每条变迁上规定一个可以触发的时间区间,此时间区间是这个状态上的局部时钟,此类模型比较适用于进行 workflow 研究^[7].但是当协议中使用多个全局时钟触发协议行为时,TPN 和时间变迁系统不能直接表述了.从 80 年代开始,很多人研究用时间自动机的形式来进行时间系统建模,文[8]的描述和方法逐渐成为标准^[9].

本文中区域自动机的定义对文[8]的定义做了改动,增加了时间变迁,并证明了区域自动机和原系统之间在外部可观测行为上是互模拟的,这样由区域自动机生成的测试例就可以通过参数化而形成可执行的测试例.

在通讯协议中,时钟约束都是表示成时钟和整数之间关系,如果不是整数而是一个有理数,可以通过选取小的时间单元,使得时间约束中只出现整数.

定义 1 (时钟)

X 是一个时钟的集合, X 上的时间约束 δ 满足

$$\delta = x \leq c \mid x \geq c \mid \neg \delta \mid \delta_1 \wedge \delta_2$$

其中 x 是 X 中的一个时钟,而 c 是一个整数常量.■

在带有时钟约束的系统中,系统的输入和输出之间是松散耦合,而不是像 Mealy 型有限状态机那样输出紧跟输入出现在一条变迁上.一条变迁上只包含一个输入事件或者一个输出事件,伴随有时钟约束和时钟复位操作.

定义 2 (时间输入输出状态机)

一个时间输入输出状态机 A 是一个五元组 $\langle \Sigma, S, S_0, C, E \rangle$,其中

Σ 是字母表,分为以“?”开头的输入消息和以“!”开头的输出消息

S 是有限状态集,

$S_0 \subseteq S$ 是初始集合,

C 是有限的时钟集合

$E \subseteq S \times S \times \Sigma \times 2^C \times \Phi(C)$ 是变迁集合.一条边 $\langle s, s', a, \lambda, \delta \rangle$ 表示一条变迁,状态 s 在事件 a 的情况下如果满足时间约束 δ ,会触发该变迁,系统进入状态 s' ,集合 $\lambda \subseteq C$ 表示这条变迁发生时需要复位的时钟.■

定义 3 (时间输入输出自动机的扩展状态)

对于一个时间输入输出自动机 $\langle \Sigma, S, S_0, C, E \rangle$,扩展状态是一个有序对 (s, v) , $s \in S, v$ 是 C 的一个解释,就是每个时

钟给定一个值, $x \in C, v(x)$ 表示时钟 x 当前的值.■

为了表述清楚,作以下约定, $\lfloor x \rfloor$ (表示 x 的整数部分, $\text{fract}(x)$ 表示 x 的小数部分.对于两个扩展状态, (s_1, v_1) 和 (s_2, v_2) , 如果 $s_1 = s_2, \lfloor v_1(x) \rfloor = \lfloor v_2(x) \rfloor$, 而且 v_1 各个时钟的小数部分的次序与 v_2 相同,那么从这两个扩展状态出发的系统的外部行为是非常相像的,它们对外的交互序列是相同的,区别仅在于发生的快慢.因为时钟的整数部分决定了系统的时钟约束是否被满足,而多个时钟的小数部分的次序决定了在这些时钟中的哪个最先改变其整数部分.如果能够定义等价关系,使得这样的扩展状态等价,并将等价状态合并,就能大大减少状态的数目.虽然时钟的整数部分可以无限大,但对某个时钟 x , 如果关于它的时间约束中最大的常数为 c , 那么对于任意 $v(x) > c$, 都是等价的.

定义 4 (时钟区域)

$A = \langle \Sigma, S, S_0, C, E \rangle$ 是一个时间输入输出自动机.对于 $x \in C$, 令 $c_x = \max \{ c \mid (x \leq c) \vee (c \leq x) \}$ 是对时钟 x 的约束.下面定义时钟解释上的等价关系, v 和 v' 是 C 上的两个解释, $v \sim v'$ 当且仅当:

(1) $\forall x \in C, \lfloor v(x) \rfloor = \lfloor v'(x) \rfloor$, 或者 $v(x) \geq c_x$, 且 $v'(x) \geq c_x$

(2) $\forall (x, y) \in C$, 如果 $v(x) \leq c_x, v(y) \leq c_y$, 那么 $\text{fract}(v(x)) \leq \text{fract}(v(y))$ 当且仅当 $\text{fract}(v'(x)) \leq \text{fract}(v'(y))$.

(3) $\forall x \in C$, 如果 $v(x) \leq c_x, \text{fract}(v(x)) = 0$ 当且仅当 $\text{fract}(v'(x)) = 0$.

A 上的一个时钟区域(clock region)是在等价条件 \sim 下的时钟解释的等价类,用 $[v]$ 表示 v 所在的时钟区域.■

对于一个有两个时钟 x 和 y 的系统, $c_x = 2, c_y = 1$, 时钟区域如图 1 表示, 共有 28 个时钟区域.

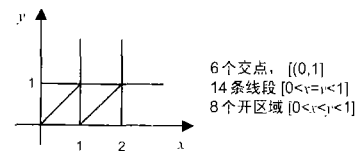


图 1 时钟区域举例

图 1 中, $[0 < x < y < 1]$ 在经过一段时间后,会离开原有的时钟区域,进入另一个时钟区域 $[0 < x < y = 1]$, 后者称为前者的时钟区域后继.

定义 5 (直接时钟区域后继) 时钟区域 α' 是 α 的直接时钟区域后继当且仅当, $\forall v \in \alpha, \exists t \in R^+, \text{满足 } v + t \in \alpha' \text{ 且 } \forall v' < t, v + t' \in \alpha$. ■

下面的定义和文[8]不同,引入了时间变迁,为每个区域设定超时事件,每个区域至多有一个超时事件.

定义 6 (区域自动机)

对于一个时间输入输出自动机 $A = \langle \Sigma, S, S_0, C, E \rangle$, 其对应的区域自动机(Region Automaton) $R(A)$ 是 $\{\Sigma \cup t\}$ 上的一个变迁表,其中 t 是时间变迁. $R(A)$ 的状态形如 $\langle s, \alpha \rangle$, 其中 x 是时钟区域.

初始状态 $\langle S_0, [v_0] \rangle$, 其中 $S_0 \in S, v_0(x) = 0, \forall x \in C$.

$R(A)$ 中有一条变迁 $\langle \langle s, \alpha \rangle, \langle s', \alpha' \rangle, a \rangle$ 当且仅当 $a \neq t$, 存在 $\langle s, s', a, \lambda, \delta \rangle \in E, \alpha$ 满足 δ , 且 $\alpha' = [\lambda - > 0] \alpha$.

$a = t, s = s'$ 并且 α' 是 α 的直接时钟区域后继. ■

定义 7 (运行和映射)

时间输入输出自动机 A 上的一次运行 $r = (\bar{S}, \bar{v})$ 具有如下形式:

$$r: \langle S_0, v_0 \rangle \xrightarrow{\sigma_1} \langle S_1, v_1 \rangle \xrightarrow{\sigma_2} \langle S_2, v_2 \rangle \xrightarrow{\sigma_3} \dots$$

它在区域自动机上的映射是 $[r] = (\bar{S}, [\bar{v}])$

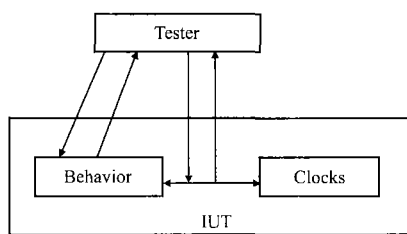
$$r: \langle S_0, [v_0] \rangle \xrightarrow{\sigma_1} \langle S_1, [v_1] \rangle \xrightarrow{\sigma_2} \langle S_2, [v_2] \rangle \xrightarrow{\sigma_3} \dots \blacksquare$$

在我们使用到的测试生成算法中,输出可以出现在允许的任意时刻,这与文[4]中假设输出在整数时刻不同.下面要证明区域自动机的每个时钟区域能够代表其所包含的每个时钟解释,区域自动机和原系统可以互模拟.

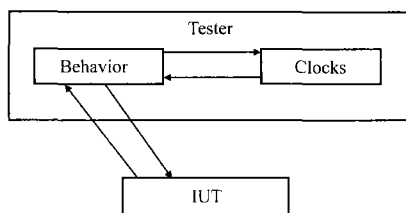
定理 1 如果 r 是 $R(A)$ 上运行的一次运行,则 A 上从 $\langle S_0, [v_0] \rangle$ 上任意一点出发,存在一次运行 r' , 满足 r 与 $[r']$ 具有相同序列,区别仅仅是时间变迁的长度不同.

证明 用构造证明 r' 的存在性.

设 $r: \langle S_0, \alpha_0 \rangle \xrightarrow{\sigma_1} \langle S_1, \alpha_1 \rangle \xrightarrow{\sigma_2} \langle S_2, \alpha_2 \rangle \xrightarrow{\sigma_3} \dots$, r' 初始格局为 $\langle S_0, v_0 \rangle$, $v_0 \in \alpha_0$, 从头开始构造 r' 的序列. 假设 r' 的第 i 个格局为 $\langle S_i, v_i \rangle$, $v_i \in \alpha_i$, 如果在 r 中当前变迁为 $\langle S_i, \alpha_i \rangle \xrightarrow{a \neq t} \langle S_{i+1}, \alpha_{i+1} \rangle$, 由区域自动机的定义可知,当前格局 $\langle S_i, \alpha_i \rangle$ 存在一条变迁 $\langle S_i, S_{i+1}, a, \lambda, \delta \rangle \in E$, α_i 满足 δ , 且 $\alpha_{i+1} = [\lambda \rightarrow 0] \alpha_i$, 因为 $v_i \in \alpha_i$, 所以 v_i 满足 δ , 令 $v_{i+1} = [\lambda \rightarrow 0] v_i$, 可知 $v_{i+1} \in \alpha_{i+1}$, 由时间输入输出自动机的定义知, $\langle S_i, v_i \rangle \xrightarrow{a \neq t} \langle S_{i+1}, v_{i+1} \rangle$.



(a)En-No98 的测试结构



(b)Petit99 的测试结构

图 2 两种跟踪时钟的测试结构

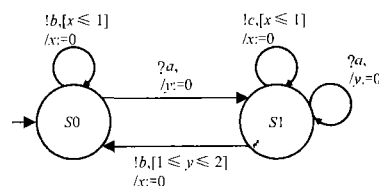


图 3 RIP 协议路由更新过程模拟

图 3 是 RIP 协议^[10,11]路由更新过程的一个模仿.当一条新路由 R_a 到达时, R_a 需要被更新到路由表中, 并且向邻居广播该更新信息. 如果周期性的收到 R_a , 那么这条路由一直有效, 反之如果在一段时间内没有收到 R_a 的更新报文, 则认为 R_a 过期, 需要从路由表中删除. 为了防止连锁反应引起网络上突发流量, 更新事件可以在一个时间段内发生. 当路由信息 R_a 到达, 需要将伴随这条路由的时钟 y 启动, y 用于记录上一次更新到现在的时间间隔, 每当收到 R_a , y 会被重置. 状态 $S1$ 在时钟 y 位于 $[1, 2]$ 区间内时既可以接受一个 $?a$ 事件留在状态 $S1$, 也可能发出一个 $!b$ 事件进入状态 $S0$. 所以在这个问题中存在同时满足事件约束的输入事件和输出事件, 确定性行为的假定不成立.

下面利用定义 6, 生成图 3 对应的区域自动机(图 4). 如前所述, 在网络协议中, 对边界值并不敏感, 可以将边界值(如 $y = 1$)并入相邻得时钟区域.

如果在 r 中当前变迁为时间变迁 t , $\langle S_i, \alpha_i \rangle \xrightarrow{a=t} \langle S_{i+1}, \alpha_{i+1} \rangle$, $S_{i+1} = S_i$, α_{i+1} 是 α_i 的直接时钟后继, 因为 $v_i \in \alpha_i$, v_i 中的时钟小数部分排列与 α_i 中相同, 那么 t' 为让 v_i 离开 α_i 的最小时间, $v_{i+1} = v_i + t'$, 则 v_{i+1} 中无论小数部分还是整数部分都与 α_{i+1} 中的元素相同, 得 $v_{i+1} \in \alpha_{i+1}$, 存在变迁 $\langle S_i, v_i \rangle \xrightarrow{a=t'} \langle S_{i+1}, v_{i+1} \rangle$, $S_{i+1} = S_i$. 反复调用上面的过程, 就能从构造出 r' , $[r']$ 具有与 r 相同的序列, 仅仅是时间变迁的长度不同. \blacksquare

有了区域自动机, 就可以将全局的时间约束转换成局部的时间约束, 将时间约束转换成时间变迁, 就可以用现有的测试方法生成抽象测试例.

3 网络协议中时间约束的测试

要测试系统的时钟, 需要有效的方法来跟踪被测系统的内部时钟. 我们采用文[3]的测试结构(图 2(b))而不是文[2]的结构(图 2(a)), 因为被测系统作为一个黑盒, 其内部时钟是无法观测的.

已有的工作都要求系统的行为是确定性的, 规定在当前时刻最多只能有一个事件是可以被触发的. 但是在实际的协议中, 为了避免网络上突发流量, 协议对于某些内部时钟加一个随机数, 使得时钟触发事件变成一个满足一定约束的概率事件, 在这种情形下, 输入输出事件的条件约束有可能同时被满足.

测试者可以为每个系统时钟设定一个跟踪时钟, 这样系统可以向跟踪时钟询问到当前时刻. TTCN 只定义局部时钟, 测试例需要知道当前哪个时钟最快到达下一个时钟区域. 为此定义一个测试集变量二元组 (x_0, y_0) , 用于记录进入该区域时的时钟值, 这里有一个隐含操作, 就是每次一条变迁发生时, 进入一个区域的时候 Tester 都要向跟踪时钟查询当前各个时钟的数值. 应用图 2(b)的测试结构, 要记录 (x_0, y_0) 的值是很容易的. 记录进入一个区域时的时钟值, 可以计算在这个区域最长的逗留时间, 就能够得到测试例的参数. 当一条变迁发生而某些时钟需要复 0 时, 需要向跟踪时钟发出复 0 信号, 例如在区域 $K0$ 上发生 $!b$ 事件时需要将时钟 x 置成 0.

在这个例子的测试中, 不允许 $x > 1$ 或者 $y > 2$ 这样的区域, 用一个 fault 区域来代替. 进入 fault 区域则表示被测实现出错, 这是时钟超时情况下的状态错. 图 4 中 e 表示一个很小的时间增量, 表示已经超过了既定的时间限制. 这样就为每个

这样得到的测试例中时间输入 T 还没有参数化,所以只是抽象测试例,需要在测试执行的时候参数化.下面以 $a.a.T.T$ 为例说明测试例的生成过程.图 5 是 $a.a.T.T$ 对应的行为树. $a.a.T.T$ 作为输入的测试例,通过前两个输入 $a.a$,系统进入状态 $K1$.此时输入第一个时间变迁 T ,参数为 $t = 1 - x0 + e$.如果 timeout,则发现错误,结束;如果收到输出 c ,则进入状态 $K2$.第二个 T 的参数为 $t = 1 - y0$,如果输出为 c ,则当前状态仍为 $K2$;如果这个时钟 timeout 而没有输出,则系统进入状态 $K3$.

表 2 $a.a.T.T$ 对应的 TTCN 测试例

Test Case Dynamic Behavior				
Test Case Name:	RIP111			
Group:	RIP/RouterUpdate/			
Purpose:	Test the timer restriction in RIP route update			
Default:	NULL			
Comments:	NULL			
No	Behavior Description	Label	Cons Ref	Verdict
1	Test Prefix			
2	!a			
3	!a			
4	Starttimer($T1 = 1 - x0 + e$)			
5	?c			
6	Starttimer($T2 = 1 - y0$)			
7	?c			PASS
8	?otherwise			FAIL
9	timeout T2			PASS
10	?otherwise			FAIL
11	timeout T1			FAIL
12				

Extended Comments: NULL

将这个测试例转换成 TTCN 形式的测试例,如表 2 所示.被测系统收到消息相当于测试系统向被测系统发送一条消息,TTCN 描述的是测试器的行为,所以测试例中的收发和上面行为树是相反的.

这条测试例对应的行为树有 2 种正确的可能分支,对非确定性自动机中,存在着不可控制的情形,从测试行为树本身不能说明行为树的哪个分支会被遍历.在真正的协议中,如果非确定性是随机出现的,那么重复有限次该测试,应该能覆盖每一种情形.这是非确定性自动机测试中的完全覆盖假设^[12].

5 错误模型

错误覆盖研究测试集可以发现被测实现中的可能错误的的能力.测试集一般是基于某种测试生成算法构造出来的,可以

按照一定的方式分类,如状态覆盖、变迁覆盖等.错误模型(fault model)规定了系统实现中可能出现的各种错误.文[13]将错误分为变迁/输出错误(Transfer/Output faults)和时间错误(Timing faults),其中时间错误可以细分为:

(1)时钟重置错误:没有按照协议规定重置时钟

(2)时钟约束范围缩小错误:如时间约束为 $1 < x < 3$,而实现中为 $1 < x < 2$,这种错误只发生在输入事件中,因为在输出只发生在 $1 < x < 2$ 是符合协议规定的,输入则因为其不允许在 $2 < x < 3$ 区间内发生被认为错误.

(3)时钟约束范围放大错误:如时间约束为 $1 < x < 3$,而实现中为 $1 < x < 5$,这种错误在输入/输出事件中都会发生.

定理 2 在非确定性自动机测试中的完全覆盖假设下,时钟约束错误能够被用 W_p 方法生成的测试集检测出来.

证明思路是按照上面几种错误类型分别检验.由于发生错误以后会进入与规定不同的时钟区域或者状态,而 W_p 方法是用特征序列来验证系统的状态和变迁,在完全覆盖的假设下,所有可能的变迁都会被遍历,这样就能够发现系统是否进入了错误的时钟区域.■

6 结论和展望

时间约束在通讯协议中普遍存在,对协议实现的时间约束进行测试,是验证协议实现的实时特性的重要方法.本文考查了当前实时系统测试方面的进展,结合路由协议测试的实践,修正了现有方法几个不符合实际需要的假设,扩展了时间自动机的理论和思路,提出了测试通讯协议时间约束的测试生成方法,并以 RIP 协议的时间约束为例说明了生成基于 TTCN 测试例的方法.

本文解决了两个理论问题,一是将全局的系统时钟转换成变迁上的局部时钟,另一个是带参数的时间变迁在不确定性情况下的测试生成.本文是结合实际工作需要提出的方法,还需要在今后的工作中更多地考查各种具有实时特性的系统的测试问题,以进一步改进现有的方法.考虑到时间自动机具有较高的复杂度,需要尽量降低复杂度,生成简短有效的测试例.时间 Petri 网的变迁具有局部时钟,而且 Petri 网模型内在的具有并发性^[7],所以进一步工作可以考虑运用 Petri 网来解决类似问题,比较不同方法的优势与不足.

在错误模型中提到时钟约束范围扩大和缩小的问题,在一些协议中,具体的时间参数是用户自定义的,时钟范围是预先不知道的.在这种情况下,需要进行时钟约束范围的测量,这种测量带有概率统计意义,是时间约束系统研究的一个方向.

参考文献:

[1] Misic V, et al. Towards a framework for testing distributed multimedia software systems[A]. International Symposium on Software Engineering for Parallel and Distributed Systems PDSE98 [C]. Kyoto: IEEE Computer Society Press, 1998. 72 - 82.

[2] En-Nouary A, et al. Timed test cases generation based on state characterization technique[A]. Real-Time Systems Symposium[C]. Madrid,

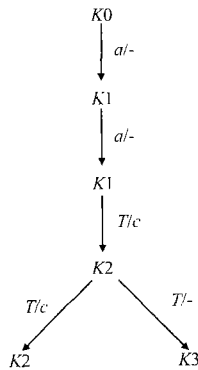


图 5 $a.a.T.T$ 对应的行为树

- Spain: IEEE Computer Society Press, 1998. 220 - 229.
- [3] Petitjean E, Fouchal H. A realistic architecture for timed testing [A]. Engineering of Complex Computer Systems [C]. Las Vegas: IEEE Computer Society Press, 1999. 109 - 118.
- [4] Springintveld J, et al. Testing timed automata [J]. Theoretical Computer Science, 2001, 254(1 - 2): 225 - 257.
- [5] Yao Y. A Petri net model for temporal knowledge representation and reasoning [J]. IEEE Trans Systems, Man, and Cybernetics, 1994, 24(9): 1374 - 1382.
- [6] Ostroff J. Real-time temporal logic decision procedures [A]. Real Time Systems Symposium [C]. Santa Monica, Ca: IEEE Computer Society Press, 1989. 92 - 101.
- [7] 刘婷, 林闯, 刘卫东. 基于时间 Petri 网的工作流系统模型的线性推理 [J]. 电子学报, 2002, 30(2): 245 - 248.
- [8] Alur R, Dill D L. A theory of timed automata [J]. Theoretical Computer Science, 1994, 126(2): 183 - 236.
- [9] Edmund M, et al. Model Checking [M]. Cambridge, Mass: MIT Press, 1999.
- [10] RFC 1058, Routing Information Protocol [S]. 1988.
- [11] RFC 2453, RIP Version 2 [S]. 1998.
- [12] Luo G, et al. Test selection based on communicating nondeterministic finite-state machines using a generalized Wp-method [J]. IEEE Trans on Software Engineering, 1994, 20(2): 149 - 162.

- [13] En-Nouary A, et al. Fault coverage in testing real-time systems [A]. The Sixth International Conference on Real-Time Computing Systems and Applications [C]. Hong-Kong: IEEE Computer Society Press, 1999. 150 - 157.

作者简介:



陈东洛 男, 1976 年 9 月出生于广东省新会县, 1995 年考入清华大学计算机系, 1999 年获学士学位, 并继续攻读博士学位, 专业方向是计算机体系结构, 计算机网络协议测试, 当前工作重点是被动测试、路由协议测试。



尹霞 女, 1972 年 4 月出生于天津。现为清华大学计算机科学与工程系讲师。1995, 1997, 2000 年, 于清华大学计算机科学与工程系, 分别获得工学学士、硕士、博士学位。自 1998 年始, 作为项目负责人独立承担了国家 863、国家自然科学基金等多个项目。目前感兴趣的研究方向为形式化技术、计算机网络及其应用和协议测试。