广西科学院学报
Journal of Guangxi Academy of Sciences

2010,26(4):465~468
Vol. 26, No. 4 November 2010

基于 UPPAAL 的简单网络支付协议形式化验证*
Formal Verification of SNPP Based on UPPAAL

余兴超,马争先,王玉斌,董荣胜
YU Xing-chao, MA Zheng-xian, WANG Yu-bin, DONG Rong-sheng

(桂林电子科技大学计算机科学与工程学院,广西桂林 541004)
(School of Computer Science and Engineering, Guilin University of Electronic Technology, Guilin, Guangxi, 541004, China)

摘要:分析简单支付协议中不同银行间的交易行为和各主体的超时约束,建立消费者、商家、银行和超时计时器的时间自动机模型,并用 UPPAAL 工具验证其是否满足商品原子性。新模型在原模型的基础上,增加了超时计时器进程来负责接收来自其它进程的超时信息,在各主体的计时器触发超时之后,计时器将发送超时信息,再通过外部的仲裁程序来解决纠纷。新模型能够满足货币原子性和商品原子性,并且比原模型更加符合协议运行的实际环境。

关键词:时间自动机 电子商务协议 UPPAAL 原子性
中图法分类号:TP301.1 **文献标识码**:A **文章编号**:1002-7378(2010)04-0465-04

Abstract: The transaction actions between different banks and the overtime of main bodies in Simple Network Payment Protocol are analyzed. The timed automata models of customer, merchant, banks and overtime timer are established. UPPAAL is used to verify the satisfication of protocol with goods atomicity. The new model, based on the original one, adds overtime timer to receive the overtime informations from other processes and sends out overtime messages after overtime timer of each main bodies are triggered. Then the issue is solved by external arbitration procedure. The new model satisfies money and goods atomicity, and is more suitable than the original for the protocol in realistic environment.

Key words: timed automata, e-commerce protocol, UPPAAL, atomicity

电子商务由消费者(customer)、商家(merchant)和银行(bank)及所信任的第三方认证机构(CA)之间的信息流、资金流和物流的交互关系,各方通过遍及全球的、开放的但不安全的 Internet 相互联系[1]。在电子商务模型中,各个主体(消费者、商家、银行等)进行交互必须遵循一定的规则,以保证各方的利益和安全,并且能控制交易风险,这就是协议的模式。安全的电子商务协议是保证电子商务活动正常开展的前提。我们可以用形式化的方法来描述与验证网络协议,从而发现协议中潜在的错误。现在已经开发出多种基于时间自动机[2]的网络协议验证工具,UPPAAL[3]就是很好的工具之一。

简单网络支付协议(SNPP)[4]是由 MIT 计算机科学实验室的 Semyon Dukach 提出的。该协议使用 UDP 数据报传输,DES 对称加密技术,HOLD 技术和相互独立机制为不信任的双方实现安全支付交易。文献[5]对 SNPP 协议进行过分析和检验,但是他们仅仅给出了简化的 SNPP 协议模型,并且在给出的银行模型上还存在一些不足之处。本文在原简单网络支付协议模型的基础上,从货币原子性和商品原子性这两个角度对协议进行分析,建立了消费者、商家、银行和超时计时器的时间自动机模型,并使用 UPPAAL 工具对模型进行了验证。

收稿日期:2010-08-06
修回日期:2010-08-21
作者简介:余兴超(1987-),男,硕士研究生,主要从事自动机理论和博弈论的研究。
*广西自然科学基金项目(桂科自 0991242)资助。

1 基本概念

定义1[2] 时钟约束的集合 $\Phi(x)$ 定义为 $\delta:=x\leqslant c \mid c\leqslant x \mid \neg\delta \mid \delta_1\wedge\delta_2$，其中，$x$ 是一个时钟变量，c 是有理数集 Q 中的一个常量。

定义2[2] 时钟集合 X 的一个时钟解释 v 给每个时钟分配一个实数值，即它是一个从 x 到非负实数集 R 的一个映射。一个时钟解释 v 满足 X 上的一个时钟约束 φ，当且仅当按照 v 给出的值，φ 为真。

定义3[2] 时间自动机 A 是一个六元组 $\langle L, L_0, \sum, X, I, E\rangle$，其中，$L$ 是一个有穷位置集合；$L_0\subseteq L$ 是一个初始位置集合；$\sum$ 是一个有穷标记集合；X 是一个有穷时钟集合；I 是一个映射，它给每个位置 L 指定 $\varphi(X)$ 中一些时钟约束；$E\subseteq L\times\Sigma\times\Phi(X)\times 2^X\times L$ 是一个迁移集合，每条迁移 (l,a,δ,Y,l') 表示在满足约束条件 a 时，通过标记 δ 的迁移，位置 l 可以迁移到位置 l'。同时时钟 $Y(Y\subseteq X)$ 被重置为0。a 称为迁移的约束条件。

原子性是电子商务协议的基本性质之一，Tygar TD[6]将原子性分为3级：货币原子性、商品原子性和确认发送原子性。货币原子性为电子商务中的资金流守恒，即资金在电子商务有关各方的转移中既不会创生也不会消失。商品原子性：首先，满足商品原子性的协议一定是满足货币原子性的；其次，必须保证购买者如果付了款就一定会得到商品，购买者如果得到了商品则一定付了款。

2 简单网络支付协议

消费者首先选择信任的银行开户存入现金，得到相应的银行账号和采用对称加密产生的密钥。银行记录每个账户的现金类别和数量、现金预留表，密钥，最近交易号码和所有的历史交易号码列表的信息，通过账号来索引。HOLD信息包含账户，账号/银行身份匹配认证以及超时约束。当消费者发送HOLD信息给商家时，协议开始运行，然后商家把信息转送到自己的银行，若商家和消费者不属于同一个银行，则把HOLD信息转送到消费者银行。若消费者账户的资金足够，银行先将预留信息添加到消费者账户的现金预留列表，再返回一个确认信息给商家银行，否则整个HOLD将被拒绝。商家在收到自己的银行的通知后，将商品发送给消费者。同时，消费者收到商品后发送PAY信息确认付款，商家把该信息转送到商家银行，商家银行再把该信息转送到消费者银行，消费者银行通过外部程序把资金送到商家银行，商家银行将要求的资金数量转入商家账号，交易完成。

简单网络支付协议用符号进行描述如下：

(1) $C\rightarrow M: P, B_C, A_C, \{B_M, A_M, Nc, \$, \mathrm{HOLD}\}K_C$。

(2) $M\rightarrow B_M: B_C, A_C, \{B_M, A_M, N_C, \$, \mathrm{HOLD}\}K_C$。

(3) $If\ B_M\neq B_C$ then

$B_M\rightarrow B_C: A_C, \{B_M, A_M, N_C, \$, \mathrm{HOLD}\}K_C$ Else go to (5)。

(4) $B_C\rightarrow B_M: \{A_C, A_M, N_{Bc}, \$, \mathrm{HELD}\}K_{Bc}, B_M$。

(5) $B_M\rightarrow M: \{B_C, A_C, N_{Bm}, \$, \mathrm{HELD}\}K_M$。

M send out the product to C. If C is satisfied, then

(6) $C\rightarrow M: P, B_C, A_C, \{B_M, A_M, Nc, \$, \mathrm{PAY}\}K_C$。

(7) $M\rightarrow B_M: B_C, A_C, \{B_M, A_M, Nc, \$, \mathrm{PAY}\}K_C$。

(8) If $B_M\neq B_C$ then

$B_M\rightarrow B_C: A_C, \{B_M, A_M, Nc, \$, \mathrm{PAY}\}K_C$,

else go to (10)。

(9) $B_C\rightarrow B_M: \{A_C, A_M, N_C, \$, \mathrm{IOU}\}K_{Bc}, B_M$。

(10) $B_M\rightarrow M: \{B_C, A_C, Nc, \$, \mathrm{PAID}\}K_M$。

其中，C 表示消费者，M 表示商家，B_i 表示 i 的开户银行，A_i 表示 i 的账号，K_i 表示 i 的账户密钥，K_{Bi}，B_j 表示银行 i 和 j 之间的公共密钥，N_i 表示 i 的交易号码，P 表示商品信息，$\$$ 表示货币数量和类型，$\{x\}K_i$ 表示用密钥 K_i 加密的信息 x，$i\rightarrow j: x$ 表示 i 向 j 发送信息 x

3 协议形式化建模及验证

3.1 建立模型

对协议中的4个进程超时约束别分设为单位时间15,20,20,20。若消费者在收到商家的发货信息15个单位时间之后，仍未发送付款消息则发送超时消息；若商家在发送确认发货消息之后20个单位时间内未收到消费者的付款信息则发送超时信息；若商家银行在发送确认消息给商家20个单位时间之后没有收到付款消息则发送超时信息；若消费者银行在发送已预留资金20个单位时间之后未收到付款消息则认为是超时，发送超时消息。超时计时器在接收到以上进程的超时信息后，立即发送消息给外部仲裁程序。

建立模型时,假设消费者与商家的银行是不同银行。此外,在不影响协议关键性质检测的前提下,假设银行支付过程为理想过程,即不存在网络通信及其它设备故障的情况。

定义 5 个进程模块:消费者进程,商家进程,消费者银行进程,商家银行进程,超时计时器进程。其中消费者进程只与商家进程进行通讯,商家进程只与商家银行进程进行通讯,商家银行只与消费者银行进行通讯,超时计时器只负责接收来自以上进程的超时信息。

3.1.1 消费者进程模块

消费者选中商家提供的产品或者服务,然后填写订单并发送到商家,封装在订单中的 HOLD 信息也被发送到了商家。当付款信息发送到银行时,在该系统中可能有以下情况发生:

(1) 消费者的资金不足,交易取消;

(2) 消费者的资金可用,在完成第一步操作后,将变成等待收货状态。当商家接收到来自银行的确认信息时,商家发送产品或提供服务,并且通知消费者去确认他所订购的产品或者是服务,然后消费者发送付款信息,交易结束;

(3) 消费者没有收到所承诺的商品或者对于商品质量不满意,则 PAY 消息将不会产生,那么在计时器超时后将会发送超时消息通知外部的仲裁程序来解决纠纷。

消费者进程的时间自动机定义为 $C=\langle L_c, L_{c0}, \sum_c, X_c, I_c, E_c\rangle$,其中,$L_c=\{C_0,C_1,C_2,C_3,C_4,C_5\}$;$L_{c0}=\{C_0\}$;$\sum_c=\{\text{Cres},\text{no},\text{Mans},\text{timeout1},\text{PAY}\}$;$X_c=\{x\}$;$I_c=I_c(x)$;$E_c\subseteq L_c\times\Sigma_c\times\Phi(X_c)\times 2^{X_c}\times L_c$。

消费者进程模型如图 1 所示。

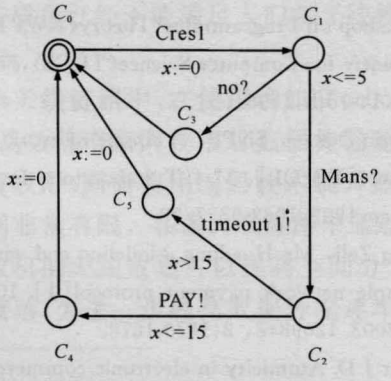

图 1 消费者时间自动机模型

3.1.2 商家进程模块

协议开始,商家进程正在等待接收来自消费者的付款信息。在该模块中有以下情况发生:

(1) 消费者资金不足,交易将取消;

(2) 当商家接收到银行的 HOLD 确认信息时,商家开始发货并提醒消费者去确认产品。商家收到消费者 PAY 的消息后,转发至自己的银行,然后商家银行在完成银行内部的资金传送后再把相应的资金划入商家的账号;

(3) 商家发送了所承诺的产品,却没有收到消费者的 PAY 消息,那么在计时器超时之后发送超时信息通知外部的仲裁程序来解决纠纷。

商家进程的时间自动机定义为 $C=\langle L_M, L_{M0}, \sum_M, X_M, I_M, E_M\rangle$,其中,$L_M=\{M_0,M_1,M_2,M_3,M_4,M_5,M_6,M_7,M_8,M_9,M_{10}\}$;$L_{M0}=\{M_0\}$;$\sum_M=\{\text{Cres},\text{Mreq1},\text{Bans1},\text{Bans2},\text{no},\text{Mans},\text{timeout2},\text{PAY},\text{Mreq2},\text{Mfund}\}$;$X_M=\{y\}$;$I_M=I_M(y)$;$E_M\subseteq L_M\times\Sigma_M\times\Phi(X_M)\times 2^{X_M}\times L_M$。

商家进程模型如图 2 所示。

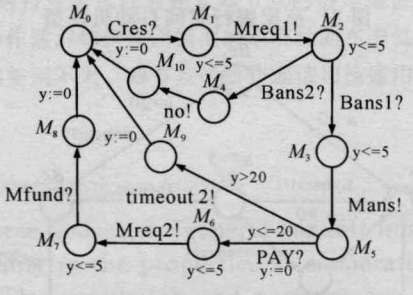

图 2 商家时间自动机模型

3.1.3 商家银行和消费者银行进程模块

商家银行接收到来自商家的付款信息时,将发送付款信息到消费者的银行去确认付款状态。有以下情况发生:

(1) 消费者的资金不足,交易取消;

(2) 若资金可用,预留请求将被确认,消费者银行将反馈"HOLD"信息给商家银行。当商家收到来自消费者的 PAY 信息,传送该信息到他的银行,商家银行再把 PAY 信息发送给消费者银行,银行之间通过内部程序完成资金传送后,商家银行将钱放入商家的账号中,并且付款操作终止;

(3)若在计时器超时仍未收到商家传递的 PAY 信息,冻结预留资金,发送超时信息。

商家银行进程的时间自动机定义为 $Bm=\langle L_{Bm}, L_{Bm0}, \sum_{Bm}, X_{Bm}, I_{Bm}, E_{Bm}\rangle$,其中 $L_{Bm}=\{Bm_0,Bm_1,Bm_2,Bm_3,Bm_4,Bm_5,Bm_6,Bm_7,Bm_8,Bm_9,Bm_{10},Bm_{11}\}$;$L_{Bm0}=\{Bm_0\}$;$\sum_{Bm}=\{\text{Mreq1},\text{Bmmes1},\text{notenough},\text{enough},\text{Bans2},\text{Bm1},\text{Bans1},\text{timeout4},\text{Mreq2},\text{Bmmes2},\text{Cfund},\text{Mfund}\}$;$X_{Bm}=\{z\}$;$I_{Bm}=I_{Bm}(z)$;$E_{Bm}\subseteq L_{Bm}\times\Sigma_{Bm}\times\Phi(X_{Bm})\times$

$2^{X_{Bm}} \times L_{Bm}$。

消费者银行进程的时间自动机定义为 $Bc = \langle L_{Bc}, L_{Bc0}, \sum_{Bc}, X_{Bc}, I_{Bc}, E_{Bc} \rangle$，其中 $L_{Bc} = \{Bc_0, Bc_1, Bc_2, Bc_3, Bc_4, Bc_5, Bc_6\}$；$L_{Bc0} = \{Bc_0\}$；$\sum_{Bc} = \{$BMmes1, enough, notenough, timeout3, BMmes2, Cfund$\}$；$X_{Bc} = \{u\}$；$I_{Bc} = I_{Bc}(u)$；$E_{Bc} \subseteq L_{Bc} \times \Sigma_{Bc} \times \Phi(X_{Bc}) \times 2^{X_{Bc}} \times L_{Bc}$。

商家和消费者银行进程模型分别如图3，图4所示。

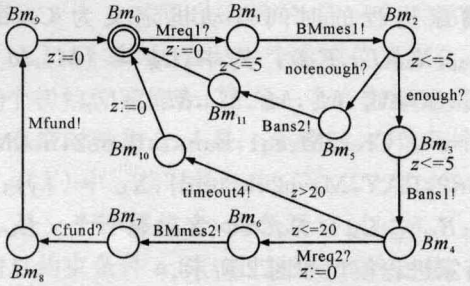

图3 商家银行时间自动机模型

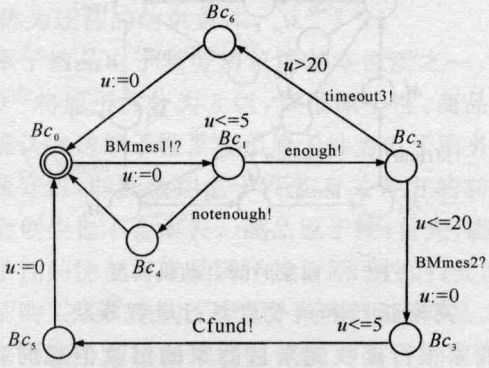

图4 消费者银行时间自动机模型

3.1.4 超时计时器进程模块

超时计时器的时间自动机定义为 $T = \langle L_T, L_{T0}, \sum_T, X_T, I_T, E_T \rangle$，其中，$L_T = \{T_0, T_1, T_2, T_3, T_4, T_5\}$；$L_{T0} = \{T_0\}$；$\sum_T = \{$timeout1, timeout2, timeout3, timeout4$\}$；$X_T = \{v\}$；$I_T = I_T(v)$；$E_T \subseteq L_T \times \Sigma_T \times \Phi(X_T) \times 2^{X_T} \times L_T$。

超时计时器进程模型如图5所示。

3.2 模型验证

应用UPPAAL工具，先对协议原模型进行验证，即验证由消费者、商家、银行进程构成的协议模型，再验证本文建立的模型。结果表明，原模型满足货币原子性，但是会存在消费者收到了商品而未付款的情况。而由协议的符号描述可知，若消费者收到了商品或服务，却对此不满意，则不会进行步骤(6)。新模型验证结果：E<>(Customer.C5 and Merchant.M9 and Bankc.Bc6 and Bankm.Bm10)满足该性质，表示消费者和商家等待信任第三方的仲裁，而银行将预留资金冻结等待信任第三方的通知。这说明通过仲裁状态可达性，使得协议的新模型满足了商品原子性。

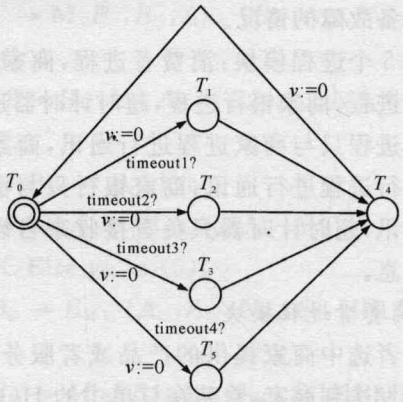

图5 超时计时器时间自动机模型

4 结束语

本文从货币原子性和商品原子性这两个角度对简单网络支付协议进行了分析，通过UPPAAL检测发现该协议原模型满足货币原子性，但在商品原子性上存在缺陷，可能产生消费者收到了商品而未付款的情况。所以在原模型的基础上，我们增加了对不同银行间交易行为和交易中超时行为的分析，设计了超时计时器模块，提出了用仲裁状态可达性来通知外部仲裁程序解决交易纠纷，从而使新模型满足了商品原子性。

参考文献：

[1] 周龙骧.电子商务协议研究综述[J].软件学报，2001，12(07)：7 1015-17.

[2] AlurR, D L Dill A. Theory of timed automata[C]. Theoretieal Computer Science, 1994, 126(2): 183-235.

[3] Aleixandre David, Wang Yi. Hierarchical timed automata for UPPAAL[C]. Presented at The 10th Nordic Workshop on Programming Theory(NWPT'98), Turku Centre for Computer Science(TUCS), Finland, October 14-16th, 1998.

[4] Semyon Dukach. SNPP: A simple network payment protocol[EB/OL]. M I T Laboratory for Computer Science, 1992, 1063-9527/92.

[5] Zhang Zeli, Ma Huadong. Modeling and verification of a simple network payment protocol[J]. 10.1109/ICCT.2003.1209848, 2: 1670-1673.

[6] Tygar J D. Atomicity in electronic commerce[C]. Proc of the 15th Annual ACM Symp on Principles of Distributed Computing, 1996: 8-22.

(责任编辑：尹 闯)