网络服务行为的进程代数验证方法研究及其应用

陈 福[1),2)] 杨家海[2)] 杨 扬[3)] 王元卓[4)] 贾美英[3)]

[1)](北京外国语大学计算机科学与技术系 北京 100089)

[2)](清华大学信息科学与技术国家实验室 北京 100084)

[3)](北京科技大学计算机科学与技术系 北京 100083)

[4)](中国科学院计算技术研究所网络重点实验室 北京 100083)

摘 要 通过形式化建模分析了系统特点,确认系统行为,从而尽可能避免系统的冲突等情况.文中提出了一种使用进程代数描述网络服务组件的行为建模方法,包括强模拟、强互模拟、时效性、触发器、服务环境等网络服务交互行为描述方法,然后给出了 P2P 节点的节点发现、防火墙穿越等行为的描述,提出了适用于托管的 RCMMS 网络自管理协议,并使用代数的方法描述协议行为,确认其合理性,从而验证文中所提出的方法.

关键词 进程代数;π 演算;服务行为;交互系统演算

中图法分类号 TP301 **DOI 号**: 10.3724/SP.J.1016.2011.01660

A Study on Network Service Behavior Verification with Process Algebra and Its Application

CHEN Fu[1),2)] YANG Jia-Hai[2)] YANG Yang[3)] WANG Yuan-Zhuo[4)] JIA Mei-Ying[3)]

[1)](*Department of Computer Science and Technology, Beijing Foreign Studies University, Beijing* 100089)

[2)](*National Laboratory for Information Science and Technology, Tsinghua University, Beijing* 100084)

[3)](*Department of Computer Science and Technology, University of Science & Technology Beijing, Beijing* 100083)

[4)](*Network Technology Research Center, Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100083)

Abstract With formalization and modeling system, we can analyze system characteristics, verify system behavior and try to avoid conflicts through. This paper presents a behavior modeling method for network services components using process algebra, including strong simulation, strong bisimulation, timeliness, service triggers, and service content. Then the P2P discovery of nodes and firewall traversal behavior description are presented. And a network self-management protocol named RCMMS is proposed with process algebra description to verify the proposed method.

Keywords process algebra; π calculus; service behavior; interactive systems calculus

1 引 言

近年来,随着物联网、云计算、三网合一等快速发展,基于互联网环境的软件系统的主要形态,开发、运行和使用方式发生了巨大变化.网络服务作为基于网络的信息处理能力的一种抽象形式得到了广泛关注.面向服务的计算(SOC)、面向服务的体系结

收稿日期:2009-05-08;最终修改稿收到日期:2011-07-25.本课题得到国家"八六三"高技术研究发展计划项目基金(2009AA01Z251,2008AA01A303)、国家"九七三"重点基础研究发展规划项目基金(2009CB320505)、国家科技支撑项目(2008BAH37B05)、国家自然科学基金(60873192,61070182,60873193,60803123)、中央高校基本科研业务费专项资金(2010XJ025)资助.**陈 福**,男,1973 年生,博士,讲师,主要研究方向为云计算、下一代互联网管理、进程代数等.E-mail: chenfu@servst.com; chenfu@cernet.edu.cn. **杨家海**,男,1966 年生,博士,教授,博士生导师,研究领域为网络管理、网络行为学. **杨 扬**,男,1955 年生,博士,教授,博士生导师,研究领域为图像识别、网格计算等. **王元卓**,博士,副研究员,研究方向为 Petri 网、网格计算. **贾美英**,博士,研究方向为云计算、大型网络管理.

构(SOA)、基础设施即服务(IaaS)、平台即服务(PaaS)、软件即服务(SaaS)等研究受到高度重视.这种以服务及其组合为基础的系统具有并发、异构、交互的特点.目前缺乏普适的、广为接受的针对服务及其合成活动抽象本质的、形式化的表述和验证方法.它对以交互为基础的网络通信或管理协议的行为分析、验证和确认具有重要意义.基于此,本文提出了一种网络协议行为建模和分析方法,使用强模拟、强互模拟、时效性、触发器、组件情境等描述网络协议交互行为,给出了 P2P 节点的节点发现、防火墙穿越等交互行为描述,并完善 MMS 协议提出了 RCMMS 适应于托管的自管理协议,然后使用进程代数给出其协议行为的交互过程验证,从而验证本文所提出的方法.

2　组件行为演算

2.1　组件行为语义描述

网络服务组件、协议及交互性系统,需要严谨的行为定义、描述和分析,确认相应的系统行为是否符合设计要求是一个重要的问题.特别是交互行为可能传递节点、链接或通信信道,从而使得系统结构动态变化,这种交互式组件的交互、协同、组合过程具有明显的异构性、交互性、并发、并行等特点.如何根据协议或网络组件行为的逻辑规则,对交互行为进行严谨的刻画、演绎和行为推理,使组件行为功能明确、可信,从而增强系统的可信性和可管理性,为用户提供平台独立、透明、稳定、可信的系统,是网络协议和交互式系统设计的核心问题之一.

代数的方法可用于描述交互式组件、网络协议行为,分析行为可能存在的混或、竞争等形成的不确定性,具有严谨确认交互式组件行为是否符合设计要求的能力.同时由于代数演算的推理功能、行为等价相关概念,使得协议组件的取代、替换和重组能力大为增强.特别是基于互联网的组件,进行必要的行为推理和预测是非常重要的.协议组件之间的交互行为对性能具有重要影响.可见使用代数的方法刻画协议组件行为在组合、发现、确认、替代等方面具有重要作用.而使用代数的方法形式化描述计算过程也是经典的做法.

形式语义学以数学为工具,运用符号和公式对计算模型、过程和计算机程序进行严谨的描述、解释和推理,使语义形式化,籍此研究协议行为分析、计算模型、程序设计语言语义表述问题.形式语义学一般分为操作语义学[1]、指称语义学[2]、公理语义学[3]和代数语义学[4]四类.本文应用并发系统的代数语义的形式化表述能力,并结合网络协议涉及的节点、链接和运行环境高度的动态性和交互性,尤其关注具有描述移动、并发和交互功能的代数语义研究.

"并行"是利用多个处理机或其它功能部件同时工作以提高系统性能或可靠性,冯·诺伊曼在 20 世纪 40 年代提出细胞自动机可认为是并行计算思想的开端.并发概念由 Petri Carl Adam[5] 于 1962 年首先严格定义并建立了模型. Milner Robin 把可以按任意次序在系统内发生的两个事件定义为并发事件.由于并发系统本身具有的高度动态、并发、不确定、交互性和异构性,分析和研究这些特点是保证系统的健壮性、可靠性的重要基础.对具有这种特点的系统结构、行为进行严谨的数学描述,可以准确地分析系统的安全性、活性、可生存性、可管理性,对系统性能评价亦有重要意义.进程代数[6-7]是一种较为常用的刻画并发系统的数学工具.进程是系统的行为,代数是使用代数的或公理的方法研究行为.进程代数是使用代数公理研究并发、分布式、交互系统的理论[8].Baeten 定义了可称为进程代数的条件:至少包含选择、顺序、并发三个算子,并满足交互性、选择合成、幂等性、右分配性、顺序合成结合性、并行合成交互性、并行合成结合性这 7 条规则[7]. Milner Robin 从 1973～1980 年一直从事并发理论研究,并于 1980 年提出了 CCS(Calculus of Communicating Systems)[4]. CCS 的提出标志着具有完善的同余集合和语义模型的进程代数的建立.随后由 Park David[9] 提出的互模拟公理极大提高了进程代数的理论完整性,并成为进程代数中的重要概念. Hoare Tony[10-11] 提出的 CSP(Communication Sequential Process)去掉了全局变量,使用消息传递的通信模式.由于 Milner Robin 等均使用进程演算(process calculus),进程代数(process algebra)一词的提出和明确定义最早见于 1982 年 Bergstra 和 Klop[12] 的进程行为的不动点描述.随后 Bergstra、Keller 定义了通信功能从而建立了 ACP(Algebra of Communicating Process)[13-14]. 其它代表性工作包括 ISO 的 LOTOS(Language of Temporal Ordering Specifications)[15-16]、ATP(Algebraic Theory of Processes)[17] 等.使用代数演算把系统的事件或进程与时间、触发概率等信息相关联起来,从而可对并发系统定量描述或评价[18]. 进程代数在我国也得到了充分的重视,林惠民院士设计并实现了进程代数验证工具 PAM/VPAM,并

与英国 Hennessy 教授合作提出，独立发展了“符号互模拟”理论，解决了传统并发计算模型对大量实际应用不能有效模拟的问题[19-20]，在国内外产生了广泛的影响.

任何进程语言都需要定义其操作语义，对系统组件的行为效果进行描述，从而使得其所描述的语义明确、避免二义性. Plotkin 提出了一种称为结构化操作语义（Structural Operational Semantics，SOC）的语义描述方法，它把复合成分的操作语义归结为其各个组成部分的操作语义，并定义了形如 $p \xrightarrow{\alpha} p'$ 的进程变迁，表示进程 P 执行动作 α 而演化为进程 p'[21]. 并发系统的行为涉及进程的状态、组件的活动和触发活动达到某种状态的事件，对三者之间关系进行描述主要有标号变迁系统[22]、事件结构[23-24]描述等，其中事件结构描述事件之间的关系，而标号变迁系统则侧重于描述状态之间的关系. λ 演算作为顺序、函数式、读写程序的计算模型，CSP、CCS 作为静态并发、并行的计算模型，π 演算[25-27]、环境（Ambient）演算[28]、Seal[29] 演算等作为移动通信、动态并发、并行计算模型，从这个发展可知计算行为的数学描述从顺序、并行、并发到移动、交互逐步演进，而移动、并发计算行为的基础是并行计算行为，并行计算的形式语义学及形式化方法的研究是理论计算机研究中的重大问题之一. 其中 Milner 等提出的描述移动通信系统、刻画进程的互模拟等价关系的 π 演算，得到了广泛关注.

为描述组件的行为和结构，建立组件及其交互行为的描述表达式极为重要，这也是本文的重点内容. 由于 π 演算能够描述结构不断变化的、移动的、并发的、交互的系统，同时 π 演算具有成熟的互模拟、等价性等理论，因此本文提出的组件交互行为建模理论主要基于 π 演算.

名：组件之间链接的名称.

发送/接受动作：设 a 为组件的一个动作，则 $\bar{a}$ 表示动作的发送行为，a 则直接表示接受行为.

组件结构同余：如果两个组件行为表达式 P、Q 是结构同余的，则可以用名替换的形式 $\{\vec{b}/\vec{a}\}$ 由 P 转化为 Q 或相反，记为 $P \equiv Q$.

组件私有通信信道：组件 P、Q 通过通信通道 α 进行通信，并要求任何其它组件不能通过 α 与组件 P、Q 通信，记为 $(\upsilon\,\alpha)(P|Q)$. 但通道 α 本身则可通过组件 P 或 Q 传送给其它组件，这样其它组件就可以使用通道 α，从而具有描述组件动态演变的能力.

并发：记为 $P|Q$，其中 P、Q 表示并发执行的进程或线程.

输入前缀 $c(x).P$：表示在开始组件 P 之前完成通过通信通道 c 接受的消息，使用名 x 接收传送过来的名. 这种模型用于模拟等待一个网络通信事件.

输出前缀 $\bar{c}\langle y\rangle.P$：描述了在组件 P 发生前使用通道 c 将名 y 发送出去. 这种情况或通过网络发送一个消息或是一个转向语句.

复制 $!P$：创建一个新的组件 P 的拷贝. 这主要用于模拟复制过程.

建新名 $(\upsilon\,x)P$：在组件 P 中建立常量 x，为通信通道.

不确定选择 $P+Q$：不确定选择执行组件 P、Q.

$[x=y].P$：如果名 x 与 y 相等则执行组件 P.

$T.P$：执行一个内部动作 T 后执行组件 P.

$[x\neq y].P$：如果名 x 与 y 不相等则执行组件 P.

空进程 0：已经执行完毕或停止.

综上，组件基本表达式为 $S::=c(x).P|\bar{c}\langle y\rangle.P|(\upsilon x)P|0|P|P|!P$.

2.2 组件交互行为的强模拟、强互模拟

（1）组件行为强模拟

设 (Q,T) 为一个 LTS，R 为 Q 上二元关系. 满足下列条件称 R 为 (Q,T) 的强模拟：两个组件 p、q 满足 pRq，如果 $p \xrightarrow{\alpha} p'$，则存在 $q' \in Q$，$q \xrightarrow{\alpha} q'$，且 $p'Rq'$，则我们就认为 q 模拟了 p.

（2）组件的强互模拟

关系 R 是序偶 (x,y) 对的集合，关系 R 的逆 R^{-1} 为序偶 (y,x) 对的集合. 基于组件集合 Q 的二元关系 R，如果 R 及其逆均为强模拟，则称 R 为在标号变迁系统 LTS(Q,T) 的互模拟关系，且组件 p 与 q 是强模拟或强等价，记为 $p \sim q$.

（3）关联组件的强互模拟

两个服务系统 P、Q，设 (C,T) 为一个 LTS，R 为 Q 上二元关系. 若任意两个组件 p、q 满足 $p \sim q$ 条件，我们称服务系统 P、Q 在关系 R 上强模拟.

2.3 组件结构替代分析

组件可能受环境或自身影响而导致某个节点组件失败，而不仅仅和节点本身有关，但可能很快又恢复组件功能，具有很高失效概率且又有很高失效后恢复概率的特性. 同时，不同的节点可能会提供相同或相似的组件，因此可以通过上述观察等价性得到功能相似或相同的组件，从而获得可靠的虚拟组件性能.

(1)顺序可替代行为 $S_1?S_2$. 适应的条件是当系统的性能和资源条件要求不能同时运行相同或相似组件获得高可靠组件的情况. 组件 $S_1?S_2$ 的观察等价行为等同于 S_1，当 S_1 失效时通过组件 S_2 获得组件.

(2)多元顺序可替代行为 $C?\sum_{j=1}^{n}S_j\backslash(i\neq j)$，其中 $\sum_{j=1}^{n}S_j$ 是观察行为与 S_i 相同或相似的组件集合，当 S_i 失效时从 $\sum_{j}^{n}S_j$ 中取得另外一个组件.

(3)并发竞争替代行为 $S_1|S_2$. 允许两个组件 S_1、S_2 同时并发执行，组件 $S_1|S_2$ 的观察行为与最先完成的组件行为相同.

(4)多元并发竞争替代行为 $|\sum_{j}^{n}S_j$. 多个组件同时执行取得最先执行成功的组件.

2.4　组件的过程控制与组件环境描述

时控组件 $TimeServ(t,S)$：执行组件 S 直到组件执行时间超过可容忍时间 t 而终止组件 S.

组件选择执行：根据输入名确认执行某个组件，$x(v).([v=y_1]P_1+[v=y_1]P_2\cdots)$，其中 y_i 各不相同.

组件触发器 $Exec(x)\stackrel{\text{def}}{=}x(y).\bar{y}$：通过链接 x 得到链接 y，然后激活链接 y. 该式常称为触发式，下列两种情况等价：

(1)直接执行进程 P；

(2)进程 P 加前缀 z，然后通过触发式传送 z. 如 $(z)(\bar{x}z|z.P)$ 与进程 P 等价.

组件环境 C：因为组件的内部交互、内部结构对组件的使用者隐藏，组件的等价性需使用观察等价性来描述. 而观察等价性是基于组件执行环境，因此这里给出组件环境上下文的定义. 这里[]表示“空”的含义，即等待具体的组件填入.

$$C::=[\ ]|\alpha.C|(v\,x)C|C|P|P|C|C?C|$$
$$C?\sum_{j=1}^{n}Sj\backslash(i\neq j)|TimeServ(v(t,S).$$

3　P2P 行为能力描述

网络组件行为分析和建模是验证、确认、分析和管理网络组件的重要工具. 下面给出使用代数的方法来描述 P2P 组件的行为. 分别通过 P2P 发现行为、穿越防火墙行为、节点消息路由行为、P2P 缓存等给出代数语义描述.

3.1　Peer 节点发现

P2P 节点通过聚合节点发现其它各个节点. 同 peer 表示发出请求节点，x 表示标识自身信息的唯一标识；$RenderPeer$ 表示聚合节点，$cachse$ 表示 P2P 的缓存信息. 下面给出组件行为定义：

发出请求：$\overline{ask}(x).Peer$；

接受请求：$ask(x).RenderPeer$；

响应返回：$\overline{resps}(info).RenderPeer$；

接受响应：$resps(info).Peer$；

缓存信息：$(v\,cachse)Peer$；

发现组件行为描述，包括节点的发送、聚合节点接收、聚合节点响应及其之间的交互行为：

$\overline{ask}(x).Peer\xrightarrow{ask}Peer$；

$ask(x).RenderPeer\xrightarrow{ask}RenderPeer(cache/x)$；

$resps(info).RenderPeer\xrightarrow{resps}RenderPeer(cache/info)$；

$\overline{resps}(info).Peer\xrightarrow{reps}Peer$；

$\overline{ask}(info).RenderPeer\xrightarrow{ask}RenderPeer$；

$ask(x).Peer\xrightarrow{ask}Peer(x/info)$.

3.2　P2P 穿越防火墙行为分析

P2P 节点之间如果被防火墙隔离，相互之间通信的机制可按如下行为进行：外部 $Peer$ 将消息发送给防火墙外部的路由 $Peer$，外部的路由 $Peer$ 等待防火墙内部的节点定期主动发出连接路由 $Peer$，然后将相应信息发给内部的 $Peer$. 其过程描述如下(其中 $Peer1$、$RendPeer$ 分别是防火墙内部通信的两个节点，$Peer$ 为通信节点、$RendPeer$ 为转发节点，$Peer$ 为防火墙外部转发节点)：

$!\overline{outget}(signl).Peer$；
$outget(signl).RenderPeer$；
$sigl(x).RenderPeer$；
$\overline{sigl}(x).RenderPeer$；
$sigl(x).Peer$；
$\overline{sigl}(x).Peer$；

$\overline{outget}(signl).Peer1\xrightarrow{!outget}Peer1$；
$outget(signl).RenderPeer\xrightarrow{outget}RenderPeer$；
$\overline{sigl}(info).Peer2\xrightarrow{sigl}Peer2$；
$sigl(x).RenderPeer\xrightarrow{sigl}RenderPeer(x/info)$；
$\overline{sigl}(info).RenderPeer\xrightarrow{sigl}Peer1$.

$!\overline{check}(sigl).Peer$；　$send(msg).Peer$；　$forward(x).RouerPeer$；　$push(msg).RouerPeer$；
$check(sigl).Peer$；　$\overline{send}(msg).Peer$；　$\overline{forward}(x).RouerPeer$；　$\overline{push}(msg).RouerPeer$；

$\overline{send}(msg).Peer1 \mid send(msg).RouerPeer1 \xrightarrow{send} Peer1 \mid RouerPeer1(msg)$;

$\overline{forward}(msg).RouerPeer1 \mid forward(x).RouerPeer2 \xrightarrow{forward(x)} RouerPeer1 \mid RouerPeer2(x/msg)$;

$\overline{push}(msg).RouerPeer2 \mid push(msg).Peer2 \xrightarrow{push} RouerPeer2 \mid Peer2(msg)$;

$!\overline{check}(sigl).Peer2$; $check(sigl).Peer2$;

$\overline{send}(msg).Peer1 \mid send(msg).RouerPeer \xrightarrow{send} Peer1 \mid RouerPeer(msg)$;

$\overline{push}(msg).RouerPeer \mid push(msg).Peer2 \xrightarrow{push} RouerPeer \mid Peer2(msg)$.

4 网络自管理协议验证

4.1 协议提出

网络的自管理是异地远程网络管理的基础，自管理本身也是一个非常复杂的问题. 自管理(源生管理)是网络正常运行和提供组件理想的实现手段. 如交换机、路由器等网络接入点的自我管理，包括自配置、自优化、自治愈等，是从根本上提高网络可生存性的重要保证. 自管理是网络服务的热备份、迁移、配置和性能分析等网络管理远程实现的基础和前提. 而且，这种网络自管理能力本身的可生存性要高于网元数据路由的生存能力. 因此，健壮的网络管理通信能力就成为了一个核心工作. 利用现有的网络设施和通信协议建立网络管理通信是目前的主要实现手段. 这种叠加在现有数据路由交换通道之上的管理信息传输，不可避免地受到现有网络数据通信路由协议本身的安全、可靠性和可管理性的局限而受到限制，只能以一种被动的方式实现网络的配置和管理. 网络数据路由的目的是数据的内容安全、高效的转发和传输，管理通信的目标是确保数据通信目标的实现，将性质和目的完全不同的数据路由通道和管理通信路由通道不加区分地进行转发与传输是网络管理的一个重大瓶颈. 在不明显增加设备负载的前提下，将数据通信和管理通信分别路由处理，是提高网络管理能力的重要实现手段. 本质上这是网络管理和网络通信共享相同物理基础设施基础上的逻辑分离，为从根本上提高网络管理能力提供了重要保证.

这种专用于配置和管理的管理路由通道逻辑上独立于数据路由，具有简单、安全、健壮的特征. 共享物理网络设施，建立逻辑上独立的管理路由通信通道，从异地远程管理网络，国际上提出了各种解决方案. 得到了国际上本领域专家学者的充分重视. Maltz 等人[30]提出的元管理服务层 Meta Management Service Layer(MMSL)是其中的一个代表. MMSL 通过 Component Manager(CM)和 Management Authority(MA)分别实现被管网元管理端口输出和管理操作中心. 该协议能够很好地解决网络的管理通信路由与数据路由的区别. 但该协议存在一个重要的缺点. 虽然提及了 MA 失效及多个 MA 并存的情况. 但存在一下问题：(1)没有 MA 中密钥和加密源路由的处理，一旦 MA 失效只能重新洪泛、重新生成各个节点的源路由；(2)一个 MA 失效，各个 CM 都要再更新安全源路由，原来所有缓存均失效；(3)需要重新生成 MA 与 CM 之间的公钥/私钥对. 这些都需要一定的时间，从而影响了网络管理. 这类似于计算机系统出了问题必须重新启动计算机一样. 本文在 MMS 基础上提出了 Ring Chain Meta Management Service(RCMMS)，能够在不中断管理的前提下，恢复 MA 的管理能力，原公钥/私钥不变、源路由不变、生成树不变，从而极大提高了效率. 下面给出其逻辑含义并使用进程代数证明其逻辑的正确性.

4.2 RCMMS 协议

如图 1 所示，MA 作为管理节点其重要性不言而喻. 为了保护正在启用的 MA 节点的信息，使用了链式洋葱路由. 这种路由独立于数据路由，采用自身管理路由，具体实现方法和 MMS 一致，当 MA 集合中建立起后，采用任意一点作为发出加密路由原点向另外间隔特定长度的节点发出安全源路由. 而这种出发原点节点根据一定步长逐步改变，从而确保了整个链式结构信息一致，而各个节点又无法得到全局 MA 拓扑，从而当正在启用的节点失效后能够采用任意其它节点取代，原来的生成树及其中各个节点的缓存路由仍然有效. 这样就可以极大地提高整个系统的速度、安全和稳定性.

MA 发出 path explorer 洪泛消息对 CM 进行发现，为了及时发现路由失效，需要定期地发起 path explorer 洪泛消息. 这种消息带有唯一的标识符. 当 CM 从其一接口接收到 MA 的 path explorer 洪泛消息后将自身 ID 追加在 path explorer 洪泛消

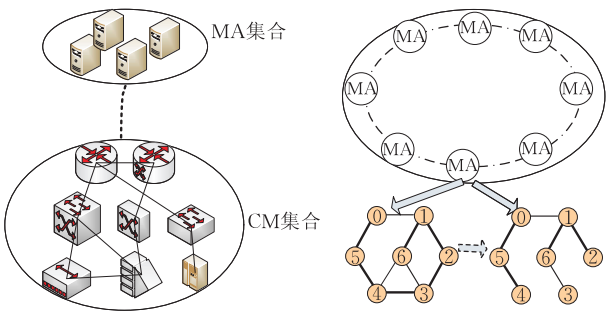

图 1　RCMMS 交互

息，然后向其自身其它接口发出，同时将该 MA 到 CM 的路径缓存用以从 CM 到 MA 逆向通信. 由于 path explorer 洪泛消息本身带有序号，这样就可以避免重复泛洪. 其行为描述为

洪泛路由：$\overline{flood}(ID).\mathrm{MA}$；$flood(ID).\mathrm{CM}$；

启动行为：$\overline{boot}(ID).\mathrm{MA}$；$boot(ID).\mathrm{CM}$；

时延路由更新洪泛：

$Timer(timerflood).\overline{flood}(pathexploer).\mathrm{MA}$；

拓扑更新洪泛：$topology.\overline{flood}(ID).\mathrm{MA}$；

缓存路由：$cache(path).\mathrm{CM}$；

链路回声：

$\overline{send}(mgmtIP).\mathrm{CM}$；$send(mgmtIP).\mathrm{MA}$；

私有通信通道：

$new(comm)(comm.\overline{comm}.\mathrm{MA}|\overline{comm}.comm.\mathrm{CM})$；

更新行为：

$update(ID).\mathrm{MA}$；$update(ID).\mathrm{CM}$；

$install(ID).\mathrm{MA}$；$install(ID).\mathrm{MA}$；

链路状态通信行为：

$\overline{send}(ID,Link).\mathrm{CM}$；$send(ID,Link).\mathrm{CM}$；

失效更新路由：

$\overline{TopologyUpdate}(flood).flood().\mathrm{CM}$；

追加路由：$append(selfID).\mathrm{CM}$；

变更路由：$change(state).Advertise(LSA).\mathrm{CM}$：

创建管理网元代理：

$create(mm0,IP).\mathrm{MA}$；$create(mm0,IP).\mathrm{CM}$；

CM 与 MA 之间通信：

$\overline{request}(x).\mathrm{CM}$，$request(x).\mathrm{MA}$；

MA 转发通信：

$forward(x).\mathrm{CM}$，$\overline{forward}(x).\mathrm{MA}$；

MA 的安全源路由：$\overline{encrypt}(PathNodes).\mathrm{MA}$；

CM 转发路由：

$encrypt(PathNodes).\overline{forward}.(PathNodes).\mathrm{CM}.$

4.3　RCMMS 协议逻辑推理

4.3.1　MA 集合节点链式洋葱路由

首次运行时，当前采用的 MA 节点将 MA 的最小生成树及各个节点的公钥信息加密. 然后当前节点 MA 将最小生成树、各个 CM 节点的公钥加密发送. 发送的终点是与当前采用的 MA 节点相隔一定的节点. 经过一定的时间间隔后该终点作为启动继续采用各个的方式进行环形路由. 当前采用的 MA 一旦受到攻击，离他最近的节点就自然成为当前节点，从而避免了相应节点的重新洪泛. 采用洋葱路由可使得各个节点对除相邻节点外的所有节点并不感知，因此极大地保护了该环路节点. 具体过程描述如下：

$receive(ID).uncover(privekey).\overline{receive}(ID).\mathrm{MA}|\overline{receive}(ID).uncover(ID)\ \overline{receive}(ID+x).\mathrm{MA}$；

$\overline{TimerServer}.\overline{receive}(ID).\mathrm{MA}|receive(ID).\mathrm{MA}.$

4.3.2　MA 集合节点链式洋葱路由

MA 节点发出洪泛或时延发出洪泛：

$\overline{flood}(ID).\mathrm{MA}|\overline{TimerServer}(timerflood).flood(pathexploer).\mathrm{MA}|topology.\overline{flood}(ID).\mathrm{MA}$；

CM 节点接受洪泛、追加 ID、缓存路径、转发洪泛，给出链路状态信息：

$flood(ID).append(selfID).\overline{send}(ID,SLA).\overline{flood}(ID).\mathrm{CM}|change(state).Advertise(LSA).\mathrm{CM}$

通过上述步骤后，一个 MA 可以获得上述各个 CM 返回的信息，从而得到网元链接的图状结构，根据 Minimum spanning tree、Dijkstra 等算法可以获得以当前 MA 为原点到各个顶点的最短路径.

当两个 CM 需要通信时，CM 通过向 MA 发送转发请求实现：

$new(request,forward)(\overline{requet}(x).\mathrm{CM1}|request(x).\overline{forward}.\mathrm{MA}|forward(x).\mathrm{CM}).$

MA 构造加密源路由：由于 MA 知道所有的 CM 密钥，因此可构建各个 CM 的安全源路由，$ConstructOnionEncryptedRoute.\mathrm{MA}$；

当 CM 发现某个邻接链路不通就会向 MA 发送链路状态通告 LSA，这是通过路由备份实现的，如最优路径、次优路径等.

4.3.3　MA 集合中 $MA1$ 替代 $MA0$

根据 RCMMS 协议要求，只要 MA 集合中的元素具有强模拟能力就可以. 这样就可以确保当前 MA 根受到攻击后能够自动的替代. 根据强模拟的定义，设关联关系 R 满足 $R=\{(MA0_0,MA1_0),(MA0_0,MA1_2),(MA0_1,MA1_1),(MA0_2,MA1_1)\}$，其关系如图 2 所示. 证明两个组件 $MA0$、$MA1$ 是强互模拟的. 其中 $MA0$ 表示正在使用的 MA 根，而 $MA1$ 则是另一具有相似功能的 MA. 如图 2 所示.

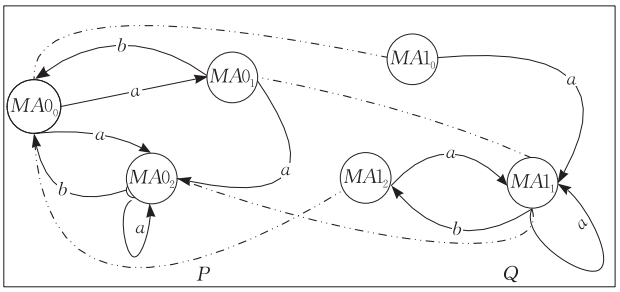

图 2 MA 组件相似备份关系

证明.

(1)因为$(P_0,Q_0)\in R$且

$MA0_0\xrightarrow{a}MA0_1$,则存在$MA1_0\xrightarrow{a}MQ1_1$,且$(MA0_1,MA1_1)\in R$;

$MA0_0\xrightarrow{a}MA0_2$,则存在$MA1_0\xrightarrow{a}MA1_1$,且$(MA0_2,MA1_1)\in R$;

(2)因为$(MA0_0,MA1_2)\in R$且

$MA0_0\xrightarrow{a}MA0_1$,则存在$MA1_2\xrightarrow{a}MA1_1$,且$(MA0_1,MA1_1)\in R$;

$MA0_0\xrightarrow{a}MA0_2$,则存在$MA1_2\xrightarrow{a}MA1_1$,且$(MA0_2,MA1_1)\in R$;

(3)因为$(MA0_1,MA1_1)\in R$且

$MA0_1\xrightarrow{a}MA0_2$,则存在$MA1_1\xrightarrow{a}MA1_1$,且$(MA0_2,MA1_1)\in R$;

(4)因为$(Ma0_2,MA1_1)\in R$且:

$MA0_2\xrightarrow{a}MA0_2$,则存在$MA1_1\xrightarrow{a}MA1_1$,且$(MA1_2,MA1_1)\in R$;

因此$MA1$强模拟$MA0$,因此完全可用$MA1$取代$MA0$.

4.3.4 MA 集合中$MA1$互替代$MA0$

如果管理组件 MA 集合中存在如下关联关系R,满足$R=\{(MA0_0,MA1_0),(MA0_0,MA1_2),(MA0_1,MA1_1),(MA0_2,MA1_1)\}$,$R^{-1}=\{(MA1_0,MA0_0),(MA1_0,MA0_2),(MA1_1,MA0_1),(MA1_2,MA0_1)\}$,其中$\{MA0_0,MA0_1,MA0_2\}\in P$,$\{MA1_0,MA1_1,MA1_2\}\in MA1$, $MA0$、$MA1$是可以互相取代的,这可以根据强等价性证明.

证明.

(1)服务系统$MA1$强模拟$MA0$.

因为$(MA0_0,MA1_0)\in R$且:

$MA0_0\xrightarrow{a}MA0_1$,则存在$MA1_0\xrightarrow{a}MA1_1$,且$(MA0_1,MA1_1)\in R$;

$MA0_0\xrightarrow{a}MA0_2$,则存在$MA1_0\xrightarrow{a}MA1_1$,且$(MA0_2,MA1_1)\in R$;

因为$(MA0_0,MA1_2)\in R$且

$MA0_0\xrightarrow{a}MA0_1$,则存在$MA1_2\xrightarrow{a}MA1_1$,且$(MA00_1,MA1_1)\in R$;

$MA0_0\xrightarrow{a}MA0_2$,则存在$MA1_2\xrightarrow{a}MA1_1$,且$(MA00_2,MA1_1)\in R$;

因为$(MA0_1,MA1_1)\in R$且:

$MA0_1\xrightarrow{a}MA0_2$,则存在$MA1_1\xrightarrow{a}MA1_1$,且$(MA00_2,MA1_1)\in R$;

因为$(MA0_2,MA1_1)\in R$且:

$MA0_2\xrightarrow{a}MA0_2$,则存在$MA1_1\xrightarrow{a}MA1_1$,且$(MA00_2,MA1_1)\in R$;

因此服务系统$MA1$强模拟$MA0$.

(2)服务系统$MA0$强模拟$MA1$.

因为$(MA1_0,MA0_0)\in R^{-1}$且:

$MA1_0\xrightarrow{a}MA1_1$,则存在$MA0_0\xrightarrow{a}MA0_1$,且$(MA1_1,MA0_1)\in R^{-1}$;

因为$(MA1_2,MA0_0)\in R^{-1}$且

$MA1_2\xrightarrow{a}MA1_1$,则存在$MA0_0\xrightarrow{a}MA0_1$,且$(MA1_1,MA0_1)\in R^{-1}$;

因为$(MA1_1,MA0_1)\in R^{-1}$且:

$MA1_1\xrightarrow{b}MA1_2$,则存在$MA0_1\xrightarrow{b}MA0_0$,且$(MA1_2,MA0_0)\in R^{-1}$;

因为$(MA1_1,MA0_2)\in R^{-1}$且:

$MA1_1\xrightarrow{a}MA1_1$,则存在$MA0_2\xrightarrow{a}MA0_2$,且$(MA1_1,MA0_2)\in R^{-1}$;

因此可知,$MA0_0$、$MA1$基于关系R强互模拟,可以相互替代.

5 结 论

通过代数演算可以充分理解计算模型、网络协议和交互行为过程,通过等价、互模拟等方法可对多网络协议行为过程进行严谨的描述和推理.目前以网络服务组件为基础的协议交互行为的数学描述研究急需加强.因此基于现有研究工作基础,加强对网络服务组件的交互行为、计算行为的行为演算研究非常必要.这对网络协议行为分析、交互过程行为定义和确认具有重要意义.

参 考 文 献

[1] McCarthy J. A basis for a mathematical theory of computation//Braffort P, Hirshberg D eds. Computer Programming and Formal Systems. Amsterdam: North-Holland, 1963: 33-70

[2] Scott D S, Strachey C. Towards a mathematical semantics for computer languages//Fox J ed. Proceedings Symposium Computers and Automata. New York: Polytechnic Institute of Brooklyn Press, 1971: 19-46

[3] Floyd R W. Assigning meanings to programs//Schwartz J T ed. Proceedings Symposium in Applied Mathematics. Mathematical Aspects of Computer Science, AMS, 1967: 19-32

[4] Milner Robin. A calculus of communicating systems//Lecture Notes in Computer Science 92. Springer Verlag, 1980

[5] Petri C A. Kommunikation mit automaten[Ph. D. dissertation]. University of Bonn, Bonn, West Germany, 1962

[6] Morris J H. Lambda-calculus models of programming languages[Ph. D. dissertation]. MIT, Cambridge, MA, USA, 1969

[7] Bekic H. Towards a Mathematical Theory of Processes. IBM Laboratory, Vienna: Technical Report TR 25.125, 1971

[8] Milne G J, Milner R. Concurrent processes and their syntax. Journal of the ACM, 1979, 26(2): 302-321

[9] Park D M. Concurrency on automata and infinite sequences//Deussen P ed. Proceedings of the Conference on Theoretical Computer Science. Lecture Notesin Computer Science 104. Springer-Verlag, 1981

[10] Hoare C A R. Communicating sequential processes. Communications of the ACM, 1978, 21(8): 666-677

[11] Hoare C A R. A model for communicating sequential processes//McKeag R M, Macnaghten A M eds. On the Construction of Programs. Cambridge, UK: Cambridge University Press, 1980: 229-254

[12] Bergstra J A, Klop J W. Fixed point semantics in process algebra. Mathematical Centre, Amsterdam: Technical Report IW 208, 1982

[13] Bergstra J A, Klop J W. ACP: A universal axiom system for process specification. CWI Newsletter, 1987, 15: 3-23

[14] Keller R M. Formal verification of parallel programs. Communication of ACM, 1976, 19(7): 371-384

[15] van Eijk P H J et al. The formal description technique LOTOS. ISO 8807, 1989

[16] Bolognesi Tand, Brinksma E. Introduction to the ISO specification language LOTOS. Computer Networks and ISDN Systems, 1987, 14(1): 25-29

[17] Hennessy M. An Algebraic Theory of Processes. Cambridge, MA: MIT Press, 1988

[18] Herzog U. Formal description, time and performance analysis: A framework. IMMD VII, Friedrich-Alexander-Universitat, Erlangen-Nurnberg, Germany: Technical Report 15/90, 1990: 172-190

[19] Lin Huimin. PAM: A process algebra manipulator//Proceedings of the 3rd Workshop on Computer Aided Verification. Heidelberg, Germany, 1993: 136-146

[20] Lin Huimin. Complete inference systems for weak dissimulation equivalences in the pi-calculus//Proceedings of the TAPSOFT. Aarhus, Denmark, 1995: 187-201

[21] Plotktin G D. A structure approach to operational semantics. Computer Science Department, Aarhus University, Aarhus, Denmark: Technical Report DAIMI FN-19, 1981

[22] Lien Y E. Study of theoretical and practical aspects of transition systems [Ph. D. dissertation]. University of California, Berkeley, 1972

[23] Nilsen M, Plotkin G D, Winskel G. Petri nets, event structure and domains, Part I. Theoretical Computer Science, 1981, 13(1): 85-108

[24] Winskel G. An introduction to event structure//LNCS 354. Berlin: Springer, 1989: 364-397

[25] Milner R. Communicating and Mobile Systems: The π-calculus. Cambridge, UK: Cambridge University Press, 1999

[26] Sangiorgi D, Walker D. The π-calculus: A Theory of Mobile Processes. Cambridge, UK: Cambridge University Press, 2001

[27] Milner R, Parrow J, Walker D. A calculus of mobile processes parts I and II. University of Edinburgh, Edinburgh, England: LFCS Report 89-85, 1989

[28] Cardelli L, Gordon A D. Mobile ambients//Proceedings of the 1st International Conference on Foundations of Software Science and Computation Structure//Nivat M ed. Lecture Notes in Computer Science 1378. Springer-Verlag, 1998: 140-155

[29] Castagna G, Vitek J, Nardelli F Z. The seal calculus. Information and Computation, 2005, 201(1): 1-54

[30] Gogineni H, Greenberg A, Maltz D A, et al. MMS: An autonomic network-layer foundation for network management. IEEE JSAC Special Issue on Recent Advances in Autonomic Communications, 2010, 28(1): 15-27

CHEN Fu, born in 1973, Ph. D., lecturer. His current research interests include cloud computing, management of NGI networks, process algebra (PI calculus, lamda calculus).

YANG Jia-Hai, born in 1966, Ph. D., professor, Ph. D. supervisor. His current research interests include computer networks, network management Internet measurement, and next generation networking.

YANG Yang, born in 1955, Ph. D., professor, Ph. D. supervisor. His current research interests cloude computing, Grid computing.

WANG Yuan-Zhuo, Ph. D., associate researcher. His current research intersets include Petri net, Grid computing.

JIA Mei-Ying, Ph. D.. Her current research interests include cloud computing, management of large-scale networks.

Background

Analysis of concurrent mobile system behavior is a very important issue in theoretical computer science. The analysis, comparison, and application of calculi are elaborated in this paper. We focus on mobile, concurrent, interactive systems calculus and compare the main logical structure, descriptive ability of π calculus in this paper. The state-of-the-art developments of mobile process algebra system are elaborated in the paper too. Finally, future research directions in the area of the mobile process algebra system are pointed out by presenting some problems on the formalization of the service-oriented systems. And at the last part of the paper we descript the interaction among the nodes in P2P, and the interaction behavior of a protocol named RCMMS.

Stochastic process calculus, interaction behavior, concurrent, parallel computing with the Pi-calculus, Extension to Pi-Calculus for Performance Evaluation are the hot topic in this field. The theory of interaction aims to provide a unified treatment to both computation models and interaction models. The starting point of the theory is to define, in a model independent manner, the two most important relationships in computer science, the expressiveness relationship between the models and the equality relationship between the objects of a model. The model independence is crucial in order to formalize the foundational postulates of computer science. The principles and the methodologies of the theory of interaction cast new lights on complete models such as the name-passing calculi and the value-passing calculi.

The National High-Tech Research and Development Plan of China under grant No. 2009AA01Z251 (New routing system and its implementation based host identification and location-addressing), No. 2008AA01A303 (New network management system based on integration of the 3 networks), the National Basic Research Program(973 Program) of China grant No. 2009CB320505 (a research on new generation of Internet architecture and protocols), Chinese Key Technology R&D Program grant No. 2008BAH37B05 (a study on new generation of trusted Internet test network), the National Natural Science Foundation of China under grant No. 60873192, No. 61070182, No. 60873193, with these projects context, we describe the behavior of service system, protocol verification, composite design et al.