

# 用PTA模型形式化分析基于Gossip协议的发布/订阅系统<sup>①</sup>

沈思铭

(同济大学 计算机科学与技术系, 上海 201804)

**摘要:** 在研究传统的发布/订阅消息中间件系统的基础之上, 结合 Gossip 协议的特点来研究发布/订阅消息中间件, 最后运用形式化方法, 通过 PRISM 仿真工具, 对仿真的模型进行形式化分析. 实验结果表明, 发布/订阅消息中间件系统的实时性受消息产生速度的影响, 在各个订阅者订阅相同消息和不同消息两种情况之下网络特性展现不同的变化, 但最终都是随着消息产生速度的增加而减小. 可靠性随着消息产生速度的增加而减小, 并且订阅者的接收缓存越大可靠性越高, 但增幅率会越来越小. 该实验模型和实验方法对于发布/订阅消息中间件系统的研究, 以及在现实环境中配置系统的相关参数有一定的帮助.

**关键词:** 发布/订阅; 概率时间自动机(PTA); Gossip 协议; 形式化分析; PRISM

## Formal Analysis of Gossip Protocol-based Publish/subscribe Systems Using the PTA Model

SHEN Si-Ming

(Department of Computer Science and Technology, Tongji University, Shanghai 201804, China)

**Abstract:** Based on the study of traditional publish/subscribe message middleware, we study the publish/subscribe message middleware with the combination of the characteristics of Gossip Protocol. Finally we use the PRISM simulation tools to formally analyze the simulation model with the formal methods. The experimental results show that the real-time performance of the publish/subscribe message middleware system is affected by the message generation rate, and under the two condition of each subscriber subscribe the same messages or different messages, the network characteristics show different changes, but ultimately decreases with the increase of the message generation speed. The reliability decreases with the increase of the message generation speed, and increases with the increase of subscriber's receive buffer, but the increase rate will become increasingly smaller. The experimental model and experimental methods will certainly help for studying the publish/subscribe message middleware system and adjusting the system parameter in the real environment.

**Key words:** publish/subscribe; probabilistic timed automata; gossip protocol; formal analysis; PRISM

发布/订阅环境中构件的分布较广, 环境因素容易表现为不确定性条件, 本文研究了不确定性环境下对发布/订阅消息中间件系统的建模和验证. 以基于概率时间自动机(PTA)对通信基础层进行建模. 在模型中, 传输信道被认为是不可靠的, 即消息在信道传输过程中可能会丢失, 且传输时延和事件代理处理消息的时

延也不可忽略.

由于发布/订阅消息中间件系统中的每个消息的接收者都是不确定的, 是由事件代理根据事件的内容及自己知道的订阅条件来发送消息给感兴趣的人. 所以基于内容的路由协议都是以某种广播协议为基础, 在此基础上再采用一些优化措施, 以避免不必要的消

<sup>①</sup> 收稿时间:2012-05-12;收到修改稿时间:2012-06-16

息转发从而提高转发效率. 为了提高路由协议的效率, 人们通常将事件代理网络的拓扑结构组织成一种树型结构或无环图结构<sup>[1,2]</sup>, 相应的基于内容的路由协议可以以这个给定的拓扑结构为基础, 采用某种策略进行高效的 消息转发. 但是, 这种路由协议的容错能力较弱, 其中任何一个节点或边的失败, 都可能会将整个网络分割成若干个不连通的子网, 而且每次向网络中增加或删除一个代理时, 都将导致很复杂的系统重配工作.

Gossip 协议的特点正好对上述的问题有一个很好的互补作用. Gossip 协议是一个通过基于与网络中的随机个成员之间的周期性数据交换模式来进行事件传输的网络协议. 这些技术是被设计用来提高具有大规模、混杂和动态特性网络的健壮性和有效性的. 因此, 随着移动 ad-hoc 网络、无线传感器网络和对等网络技术的盛行, 它们变得非常重要. 并且, Gossip 协议算法对于整个系统增加恒定的资源负载, 并不随着系统规模的扩大而增加, 所以这个系统的可扩展性很强; 它不存在单失效点, 对于链路或者进程的失效, 其可靠性只是轻微下降, 具有较强的鲁棒性; 另外简单并易于实现也是它的优势<sup>[3]</sup>.

所以, 为了研究在指定通信环境下发布/订阅消息中间件系统的可靠性和实时性, 本文拟采用 Gossip 协议作为网络节点间的通信协议<sup>[4]</sup>, 然后结合 Gossip 协议的特点, 用 PRISM 仿真工具来形式化分析模拟环境的性能, 并且针对文献[5,9]中存在的不足之处进行了进一步补充和分析.

## 1 发布/订阅消息中间件系统的网络模型

### 1.1 Gossip 路由算法

对于一个 Gossip 节点, 它拥有反熵(Anti-Entropy)和谣言传播 (Rumor-Mongering) 两种工作方式. Anti-Entropy 工作方式以固定的概率传播所有的数据, 它具有完全的容错性, 但需要较大的网络资源和 CPU 负载; Rumor-Mongering 工作方式只需要较小的网络资源和 CPU 负载, 但必须为数据定义“最新”的边界, 并且难以保证完全容错性, 对失败重启且超过“最新”时间结点的网络节点, 无法保证其最终的一致性, 需要引入额外的机制处理不一致性.

基于 Gossip 路由算法的网络环境中需要系统中的每一个节点和许多它的对等节点周期性地交换信

息. 选择与哪些对等节点通信是严格根据这些消息如何通过这个网络传播来决定的. 理论上, 每个节点都可以随机地选择这个网络中所有可用节点的一个子集. 但实际上是不可行的, 因为每个节点都要存储一张网络节点的完整成员表, 用于存储和维护的开销是非常大的.

Gossip 算法是基于冗余和随机这一主动机制来克服潜在的进程失效和链路失效, 不像传统的被动算法那样, 只有检测到失效后才重传丢失的信息. 该算法对于整个系统增加恒定的资源负载, 并不随着系统规模的扩大而增加, 所以这个系统的可扩展性很强; 它不存在单失效点, 对于链路或者进程的失效, 其可靠性只是轻微下降, 具有较强的鲁棒性; 另外简单并易于实现也是它的优势. Gossip 算法的可靠性保证被称为概率可靠, 它是完全可靠与可扩展性之间的折中处理方式. 可以通过调整三个不同的参数来获得任意高概率的可靠性<sup>[6]</sup>: (1)每个对等节点用于接收消息的输入缓存区大小 `buf_size`; (2)启动 Gossip 算法的周期大小 `time`; (3)每次随机选择的用于数据交换的对等节点的数目 `num_size`.

### 1.2 抽象的发布/订阅消息中间件系统模型

图 1 说明了一个发布/订阅系统的抽象模型, 本文将通过这个抽象模型来进行形式化检测和分析. 它包括一个发布/订阅中间件(即发布器)、组件和一些用来描述传输信道的缓存区. 每一个缓存区 `buf` 都有一个最大容量的表征 `MAX_buf`; `n_buf` 表示现在缓存区 `buf` 中的元素个数. 所有的缓存区采用先进先出的策略.

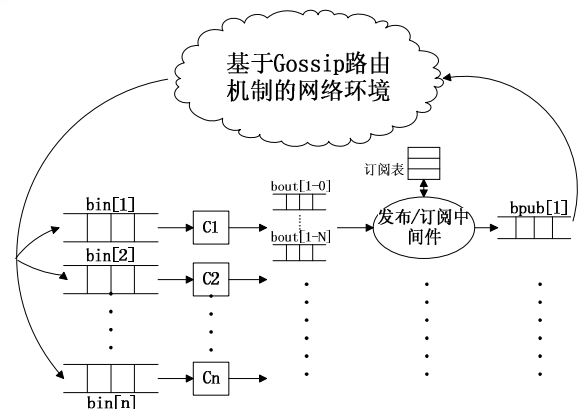


图 1 发布/订阅消息中间件系统的模型

组件要把消息(比如: 发布消息、订阅消息或者取

消订阅消息)发送给发布器, 首先把消息发送到缓存 **bout** 中, 这里使用多个发布/订阅中间件的输入缓存是为了模拟消息的优先级情况, 等级低的消息队列必须等到比它等级高的消息队列发送完之后才能发送. 根据不同等级的消息队列, 不同等级的消息相对产生速度可以和一组概率值相对应. 从发布器中发送出来的消息首先被传送到输出缓存区 **bpub** 中. 每个组件  $C_i$  都有一个输入缓存  $\text{bin}[i]$ , 它是用来接收通知消息的. 一个消息从缓存 **bpub** 到  $\text{bin}[i]$  的传输过程对应了一个从发布者到一个订阅者组件的物理链路的传输过程. 所有的消息都有序地从 **bpub** 缓存传送到目标组件的输入缓存中, 这个过程是基于储存在订阅表中的路由信息, 因为订阅表中有对相关内容感兴趣的订阅者的位置信息. 为了模拟这个不稳定的链路环境, 我们可以把从 **bpub** 到各个目标组件输入缓存的这个消息传输情况关联到一组概率值.

为了在这个抽象系统下通过模型检测技术来分析, 本文做了如下的假设: (a)传输信道保证了在其上传输的消息序列; (b)如果缓存满了, 消息就丢掉了; (c)传输信道是不可靠的. 消息在传输过程中可能根据一定的概率产生丢失的情况, 并且不同的组件之间这个概率是不一样的. 本文用  $\pi_i$  来表示第  $i$  个组件的消息丢失率. 同样论文中也引入了一些和现有系统时间参数相关的时间参量: ①**TPL**: 发布/订阅中间件(即发布器)处理一条订阅或取消订阅消息所需消耗的最小时间; ②**TPH**: 发布/订阅中间件(即发布器)处理一条订阅或取消订阅消息所需消耗的最大时间; ③**TDL**: 发布/订阅中间件(即发布器)处理一条发布消息的最小时间; ④**TDH**: 发布/订阅中间件(即发布器)处理一条发布消息的最大时间; ⑤**TCL**: 一条消息通过传输信道时所需消耗的最小传输时间; ⑥**TCH**: 一条消息通过传输信道时所需消耗的最大传输时间;

全面地分析所有这些参数, 则可能导致状态空间的爆炸. 如果我们考虑这些参数所有可能出现的组合情况, 将很容易地获得一个不好处理的确定状态模型. 对于某些参数的抽象表示可以帮助我们限制这个问题, 并且得到一个更易于分析验证的模型. 例如, 本文考虑的是同一个网络区域中的情况, 我们可以假设  $\text{TCL}=\text{TCH}=0$ , 这样一条消息通过传输信道时所需消耗的传输时间可以被忽略, 从而可能的参数值组合情况可以减少, 更便于模型的检验和分析.

## 2 概率时间自动机

概率时间自动机提供了一个模型框架, 使得与时间相关的系统和概率值相对应. 本文沿用了论文<sup>[7-10]</sup>中的定义.

一个时钟限制就是一个不确定的格式  $x \sim c$  或者是  $x \sim y \sim c$ , 其中  $\sim$  是  $\{<, \leq, >, \geq\}$  中的一个运算符.  $c$  是一个非负整数或者是一个无穷数. 一个时钟区域是一组限制条件所描述的值空间  $\mathbf{RX}$  上的一个凸子集, 其中  $\mathbf{R}^X$  表示一组时钟变量集合  $\mathbf{X}$  的时钟值的集合. 用  $Z(\mathbf{X})$  来表示集合  $\mathbf{X}$  的所有时钟区域的集合.

给定一个时钟区域  $\lambda \in Z(\mathbf{X})$  和一个评估值  $v \in \mathbf{RX}$ ,  $\lambda(v)$  是一个布尔值, 它表示把每一个时钟  $x(x \in \mathbf{X})$  替换成  $\lambda(v)$  的成功与否. 如果  $\lambda(v)=\text{true}$ , 我们可以说  $v$  满足  $\lambda$ , 用  $v \triangleright \lambda$  表示.

定义 1. 用 **AP** 表示一个原命题中一个设置好的确定集合. 一个概率时间自动机是一个 7 元组  $M=(S, S_0, L, X, \text{inv}, \text{prob}, < \tau s> s \in S)$ , 各个属性的定义如下:

①**S** 表示状态的有限集合.

②**S<sub>0</sub>** 表示初始状态.

③**L**:  $S \rightarrow 2\text{AP}$  是一个标签函数, 用于关联  $S$  中的每一个状态  $s$  和使得  $s$  有效的原子命题的集合  $L(s)$ .

④**X** 表示一个确定的时钟集合.

⑤**inv**:  $S \rightarrow Z(\mathbf{X})$  表示一个映射, 它表示一个状态和一个不变条件的映射关系.

⑥**prob**:  $S \rightarrow P_{\text{fin}}(\mu(S \times 2^X))$  是一个映射函数, 用来关联每个状态和在  $S \times 2^X$  基础之上的离散概率分布的确定非空集合.

⑦**<  $\tau s> s \in S$**  是一个函数群, 其中对于每一个  $s \in S$ ,  $\tau s: \text{prob}(s) \rightarrow Z(\mathbf{X})$  都关联了一个  $p$ ,  $p \in \text{prob}(s)$ , 并且拥有一个触发条件.

一个概率时间自动机拥有两种基本的转换类型: 推迟转换和执行转换. 这两种转换类型可以通过图 2 中的自动机来描述. 这个自动机包括两个状态  $S_0$  和  $S_1$ , 两个时钟变量  $X_0$  和  $X_1$ , 以及状态  $S_0$  和  $S_1$  之间两个状态转换过程  $a_0$  和  $b_0$ . 第一个状态转换  $a_0$  定义了一个从  $S_0$  到  $S_1$  的离散转换, 并且概率为 1. 第二个状态转换  $b_0$  定义了一个从  $S_1$  到  $S_0$  的离散转换, 并且概率也是 1.  $S_0$  是初始位置, 并且这个自动机的开始状态为  $(S_0, X_0=0, X_1=0)$ . 在时钟  $X_0$  的值达到 5 并且  $X_1$  的值达到 3 之前, 自动机可以保持在位置  $S_0$ (推迟转换). 在  $X_0$  经过 3 个时间单元并且  $X_1$  经过 2 个时间单元后, 从状态

$S_0$  到  $S_1$  的离散转换条件已经满足, 所以此时的自动机既可以转换到状态  $S_1$ (执行转换), 也可以保持在状态  $S_0$ (推迟转换).

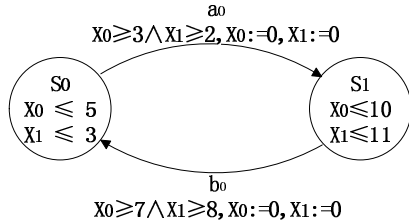


图 2 概率时间自动机的一个简单例子

### 3 发布/订阅消息中间件的形式化分析

本节将说明发布/订阅消息中间件是如何通过概率时间自动机来进行形式化分析和验证的. 一个典型的发布/订阅中间件主要由两个方面的内容: 发布器和传输信道. 接下来我们分别用概率时间自动机模型化各组成部分, 并且对 Gossip 通信网络中的网络节点进行形式化分析.

#### 3.1 发布器的形式化

图 3 说明了模型化单个发布器的概率时间自动机. 这个自动机从状态  $S_0$  开始. 触发条件和执行情况如下的转换有关:

$$\text{cond1} \wedge \text{cond2} \wedge \dots \wedge \text{condn}, \text{act1}, \text{act2}, \dots, \text{actn}$$

根据 PRISM 的帮助手册可知, 这些标签的语义是: 如果这个标签上的所有条件同时满足, 那么在这个转换终止前它将按照指定的顺序自动执行相关动作.

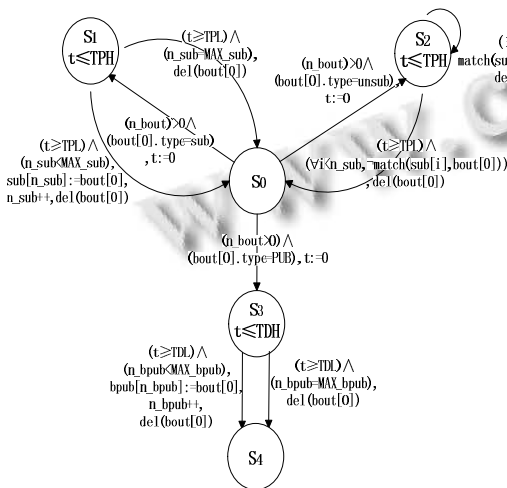


图 3 发布器的概率时间自动机模型

状态  $S_0$  是发布器的初始状态. 状态  $S_1$  是订阅状态,

该状态下发布器处理订阅类型的消息. 状态  $S_2$  是取消订阅状态, 该状态下发布器处理取消订阅类型的消息. 状态  $S_3$  是预处理状态, 该状态下发布器对接收到的消息做一个预处理. 状态  $S_4$  是发送状态, 该状态下发布器发送其输出缓存区中的消息.

发布器监视接收缓存区为  $bout$ , 如果它非空, 那么发布器接收  $bout$  缓存区中的第一条消息. 如果这条消息是订阅消息, 发布器移动到状态  $S_1$ , 同时把时间  $t$  设定为 0. 发布器可以在  $TPH$  个时间单元内保持在状态  $S_1$  中不发生转换. 在时间  $t$  经过  $TPL$  个时间单元, 并且在超时前, 发布器离开状态  $S_1$  返回到状态  $S_0$ . 此时如果订阅表没有满, 则在订阅表中记录这条订阅信息; 如果订阅表已经满了, 则把这条订阅消息丢弃掉.

如果该消息是一条取消订阅类型的消息, 那么发布器移动到状态  $S_2$ , 同时重置时间参数  $t$  为 0. 然后发布器搜索订阅表中的所有实体, 如果发现匹配项就删除. 最后, 在时间  $t$  经过  $TPL$  个时间单元, 并且在超时前, 发布器离开状态  $S_1$  返回到状态  $S_0$ . 如果是一条发布类型的消息, 发布器移动到状态  $S_3$ , 进行消息的预处理, 同时重置时间参数  $t$  为 0. 经过预处理后, 在时间参数  $t$  经过  $TDL$  个时间单元并且在未超时前, 转换到状态  $S_4$ . 在这个转换过程中需要注意的是, 如果输出缓存区  $bpub$  满了, 那么发布器就不把这条消息放入输出缓存区中了, 直接丢弃掉这条消息, 然后转换到状态  $S_4$ ; 否则发布器在  $TDL$  个时间单元的延时后, 就把该消息传送到输出缓存区  $bpub$  中.  $match$ (匹配)这个动作, 是用来模拟消息和订阅表中订阅条件的匹配情况的, 并且参数匹配时则评估为真(true). 这种方法封装了用于发布/订阅消息中间件识别和匹配的对应语言机制的详细信息.  $del$ (删除)这个动作模拟了从一个缓存区(或者是订阅表)中删除一个元素.

#### 3.2 传输信道的形式化

图 4 中的概率时间自动机模型化了发布消息的过程, 例如: 发布消息从发布器被发送到订阅组件. 此处发布者的输出缓存区假定为  $bpub$ ,  $n\_bpub$  表示缓存区中的消息个数,  $MAX\_bpub$  表示缓存区的大小; 订阅表假定为  $sub$ ,  $n\_sub$  表示订阅表中的实体个数,  $MAX\_sub$  表示订阅表的大小. 它从状态  $S_0$  开始, 并且如果缓存区  $bpub$  不为空, 则信道移动到状态  $S_1$ , 同时把变量  $i$  初始化为 0,  $i$  表示订阅表中的第  $i$  个实体的序号. 如果存在匹配项, 则发送可以该消息, 从状态  $S_1$

到状态  $S_2$  的转换模拟了这条消息和发布表中第  $i$  个实体的匹配过程. 变量  $num1$  用来记录发送订阅消息的组件的标识号. 同时变量  $num2$  用来记录此刻记录在组件接收缓存区  $bin[num1]$  中的元素个数. 状态  $S_1$  上的自循环过程说明了这个发布表中的第  $i$  个元素和这个消息不匹配, 则  $i$  的值每次都增加 1.

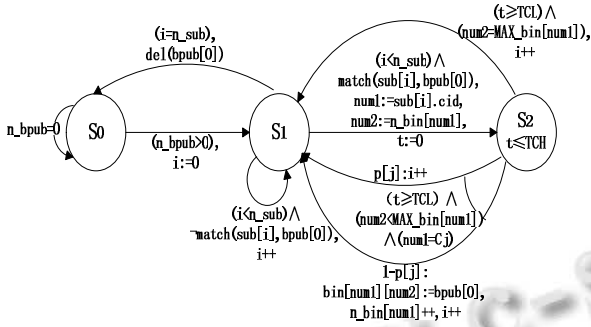


图 4 传输信道的概率时间自动机模型

### 3.3 基于 Gossip 路由机制的网络节点的形式化

图 5 中的概率时间自动机模型模拟了 Gossip 网络中网络节点的接收和转发过程. 支持 Gossip 通信协议的网络节点会周期性地触发, 进行数据的接收和转发, 文中假设每次接收的时候都是传送缓存中所有的信息, 然后接收节点会进行筛选, 最终保留下最新的消息, 并用跳数来形式化表示消息的新旧情况.

2.1 节中已经定义了这个自动机中的主要参数. 网络节点经过时间周期  $time$  个时间单元后, 首先进行接收, 如果接收消息比缓存中已有的消息新则替代掉, 否则就扔掉; 接着进行转发, 它会随机地选择  $num\_size$  个网络节点进行数据的转发, 节点在转发过程中存在数据丢失的情况, 但 Gossip 协议本身不进行被动的数据重发, 它是用基于随机和冗余的主动方式来克服潜在的数据丢失和链路失效等情况. 整个过程完成后, 该网络节点就会进入休眠期, 等待下一次的激活.

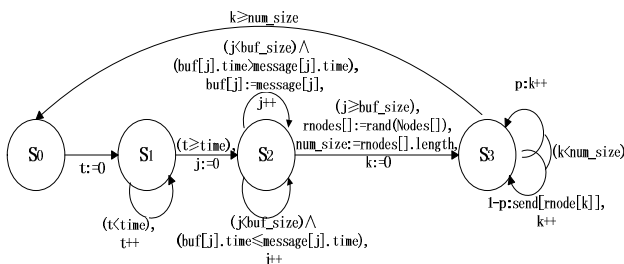


图 5 Gossip 网络中网络节点的概率时间自动机模型

## 4 模型验证及结果分析

本文选择 PRISM<sup>[11-13]</sup>模型仿真工具来验证发布/订阅系统的模型. PRISM 是英国伯明翰大学发明的一种概率时间自动机, 用来设计和分析系统, 展示其概率的特性.

因为对发布/订阅系统的概率(具有消息丢失性质的不可信信道)、稳定性和实时性的模型化感兴趣, 同时还要考虑到速率的问题, 所以本文决定采用 CTMC 这种形式化方法, 它允许融合速率这种行为类型. 从概率时间自动机到 PRISM 模型的转换是简单的. 发布/订阅系统中的每一个自动机都对应于一个 PRISM 模型. 因为所有的时钟都应该运行在同一个速率上, 所以它需要所有的执行过程在时间上都是同步的.

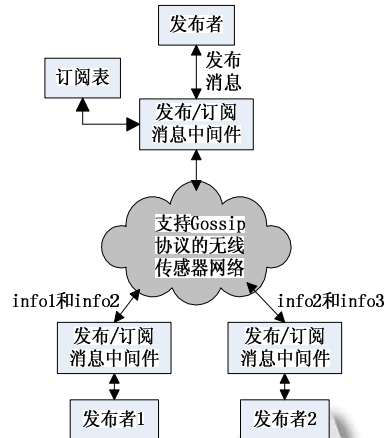


图 6 发布者与订阅者之间通信过程的模型

实验模拟了基于 Gossip 协议的发布/订阅模式的网络通信过程. 尽管用 PRISM 仿真工具相对应的形式化语言来进行形式化分析可能限制了实际情况之下的网络空间的大小, 但它同样可以反映在实际网络配置过程中可能发生的一些有参考价值的现象. 实验模型如图 6 所示, 图中总共有 1 个发布者和 2 个订阅者, 发布者发布三种消息  $info1$ 、 $info2$  和  $info3$ , 为了模拟发布消息的等级, 即先后次序, 本文假设  $info1 > info2 > info3$ , 低等级的消息必须等高等级的消息发完之后才进行发布. 订阅者 1 订阅的是  $info1$  和  $info3$  这两种消息, 订阅者 2 订阅的是  $info2$  和  $info3$  这两种消息.

实验模型由 4 个模块组成, 即: 消息产生模块、发布者模块和两个订阅者模块.

对于缓存区的形式化模式是采用 Gossip 协议本身的消息处理机制, 如果缓存满了就直接丢弃. Gossip 算

法是基于冗余和随机这一主动机制来克服潜在的进程失效和链路失效。

仿真软件运行的实验环境为双核 Intel Core i5 CPU、内存 1G×2 双通道, 操作系统 Windows XP SP3. PRISM 模型支持 rewards 结构 (即: rewards ... endrewards), rewards 结构可以描述模型中某个状态或转换所涉及到的实际值. 所以实验中我们采用 rewards 结构来分析模型的实时性和可靠性, 系统的可靠性通过消息接收率来反应, 系统的实时性通过模型的平均响应时间来反应, PRISM 可以通过该结构分析出模型中相关结构的实际值走势, 进而得出系统的相关性能指标. 但在模型中, 由于形式化语言的限制, 我们在模型中假设发布者已经拥有自身所在网络的拓扑结构, 知道接收者的地址, 同时接收者也可以接收到发布者的消息. Gossip 协议在分布式网络系统中作为一个健壮且具有可扩展性的消息传播方法已经在论文<sup>[14]</sup>中详细的介绍, 这里是建立在其基础之上进行假设和建模, 进而分析整个系统的可靠性和实时性.

第一组实验为了测试整个系统的稳定性, 主要测试消息接收率随消息产生速度的变化趋势, 通过对趋势的分析来分析系统的稳定性. 由图 7 和图 8 的实验结果可知, 当消息产生速度小于 1000packets/second 时, 整个系统的接收率比较高, 则消息的丢失率比较低, 即系统处于比较稳定的状态. 从图 7 可以看到, 当消息产生速度恒定时, 发布/订阅消息中间件系统的接收缓存区越大, 消息的接收率越高, 则消息的丢失率越低, 系统越稳定. 但可以发现, 随着缓存区大小的增加, 消息接收率的增幅在慢慢变小, 有此可知, 接收缓存区的大小不是越大越好, 其应该有最优值的. 这组仿真图对于实际应用中的消息产生速度和缓存区大小的设定具有一定的参考价值.

第二组实验是为了测试系统中消息的平均响应时间. 通过实验分析得出发布者产生消息的速度和订阅者接收缓存区大小对系统平均响应时间的影响. 两幅图都表明了, 在消息产生速度恒定的情况之下, 发布/订阅消息中间件系统的接收缓存区越大, 系统的平均响应时间就越大. 图 9 反应的是两个订阅者订阅不同类型的情况. 发现消息产生速度越快, 系统的平均响应时间就越小, 这可能是由于发布者发布同一条消息时, 如果发布给多个订阅者, 因为目的地址是不同的, 所以势必增加发布者对同一条消息的预处理次数, 从

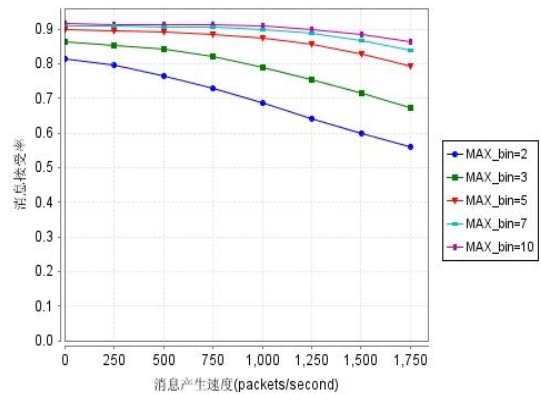


图 7 订阅者发布/订阅消息中间件的消息接收率

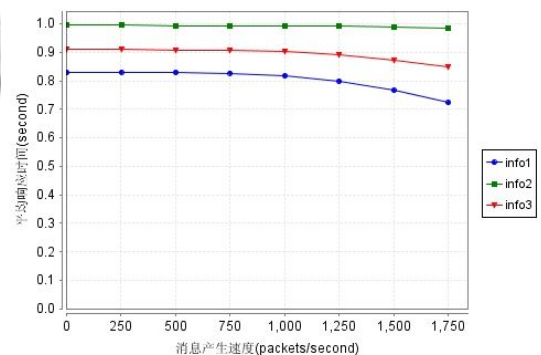


图 8 发布者发布/订阅消息中间件的消息接收率

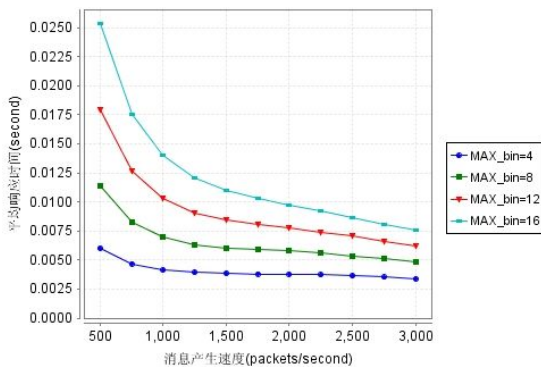


图 9 接收消息类型不一样时的消息响应时间

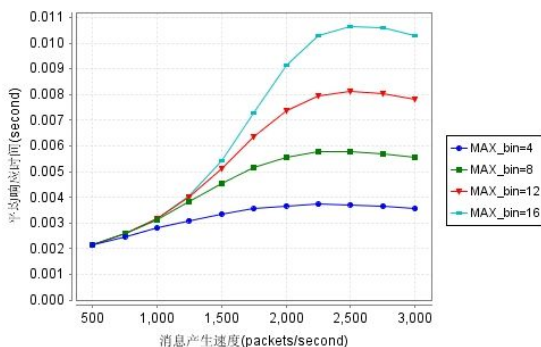


图 10 接收消息类型一样时的消息响应时间

而增加消息在队列中的等待时间,反之则相对而言减少了对同一条消息的预处理次数,减少了这条消息在队列中的等待时间.同时,从图中可以发现随着消息产生速度的增加,系统的平均响应时间渐渐趋于稳定.图10反应了消息的响应时间随着消息产生速度的增加而增加,但均在11毫秒以内,这是因为随着消息产生速度的增加,单位时间内产生的消息数量在增加,从而停留在缓冲区中的消息需要等待的时间也相应的增加了,从而导致消息的平均响应时间增加.从图中还可以看到,当消息的平均响应时间达到一个高峰后,有一个往下变小的势头,这是因为当消息产生速度超过一定的阈值时,根据队列理论可知消息的响应时间的增幅小于消息产生速度的增幅,即平均响应时间 $=(\text{info1\_waiting}+\text{info2\_waiting}+\text{info3\_waiting}+\text{in\_waiting\_1}+\text{in\_waiting\_2})/\text{send\_speed}$ ;从而使得消息的平均响应时间变小.

## 5 结论

发布/订阅模式在分布式网络环境领域中占有重要的地位,随着应用领域的渐渐拓展,发布/订阅模式势必继续为大家所关注.本文在研究现有的发布/订阅消息中间件系统和Gossip协议的基础之上,结合两者的优点,用形式化的方法进行分析.最后通过实验检验了这种机制的实时性和稳定性情况,得到了该模式下消息产生速度、消息的平均响应时间和缓冲区大小之间的关系,为该模式在实际应用中参数的设定提供了一定的参考.本文可以从网络拓扑结构方向做进一步的研究工作,可以考虑网络拓扑结构采用组播树方法时对网络传输性能的影响.

## 参考文献

- 1 Carzaniga A, Rosenblum D, Wolf A. Design and evaluation of a wide-area event notification service. *ACM Trans. on Computer Systems*, 2001,19(3):332-383.
- 2 Mastouri MA, Hasnaoui S. Performance of a Publish/Subscribe Middleware for the Realtime Distributed Control Systems. *International Journal of Computer Science and Network Security*, 2007,7(1):313-319.
- 3 薛涛,冯博琴.使用Gossip算法实现可靠的基于内容的发布订阅系统. *小型微型计算机系统*,2006,27(1):2-4.
- 4 Allavena A, Demers A, Hopcroft JE. Correctness of a gossip-based membership protocol. *Proc. of 24th ACM*

*Symposium on the Principle of Distributed Computing*. New York: ACM Press, 2005:292-301.

- 5 胡毅,于东,岳东峰,等.基于实时发布订阅模式的数控总线通信机制研究. *小型微型计算机系统*,2009,30(10):1994-1998.
- 6 Gupta I, Birman KP, Van Renesse R. Fighting Fire with Fire: Using randomized gossip to combat stochastic scalability limits. *Journal of Quality and Reliability Engineering International*, 2002,18(3):165-184.
- 7 Kwiatkowska M, Norman G, Segala R, Sproston J. Automatic verification of real-time systems with discrete probability distributions. *Theoretical Computer Science* 282, 2002: 101-150.
- 8 Kwiatkowska M, Norman G, Sproston J, Wang F. Symbolic model checking for probabilistic timed automata. Heidelberg: Springer, 2004,3253:145-150.
- 9 Kwiatkowska M, Norman G, Parker D, Sproston J. Performance analysis of probabilistic timed automata using digital clocks. *Formal Methods in System Design* 29, 2006:33-78.
- 10 He F, Baresi L, Ghezzi C, Spoletini P. Formal analysis of publish-subscribe systems by probabilistic timed automata. In: Tallinn and Estonia. FORTE 2007 27th IFIP WG 6.1 International Conference Proc., CS, Springer, 2007, 4574: 247-262.
- 11 Kwiatkowska M, Norman G, Parker D. Probabilistic symbolic model checking with PRISM: A hybrid approach. *International Journal on Software Tools for Technology Transfer (STTT)*. 2004,6(2):128-142.
- 12 Kwiatkowska M, Norman G, Parker D. Quantitative analysis with the probabilistic model checker PRISM. *Electronic Notes in Theoretical Computer Science*. 2005, 153(2):5-31.
- 13 Hinton A, Kwiatkowska M, Norman G, Parker D. PRISM: A tool for automatic verification of probabilistic systems. In: Hermanns H, Palsberg J, eds. TACAS 2006 and ETAPS 2006. Heidelberg: Springer, 2006, 3920: 441-444.
- 14 Kermarrec AM, Massoulié L, Ganesh AJ. Probabilistic reliable dissemination in large-scale systems. *IEEE Trans. on Parallel and Distributed System*, 2003,14(3): 248-258.