

11-14, 31

模糊时间 Petri 网及其时间推理方法

Fuzzy time Petri Nets and Temporal Reasoning Method

高梅梅 吴智铭

TP18

(上海交通大学电子信息学院自动化系 上海 200030)

Abstract The paper presents a translation from Petri nets to linear logic with the objective of enhancing the analysis ability of Petri nets. The definition of Petri nets is described by defining the translations in Petri nets using the connectives in linear logic, and the behavior of Petri nets is constructed using the rules we defined. A fuzzy time Petri net model and its temporal reasoning method using linear logic are proposed in the paper. The behavior of Petri nets can be analyzed precisely and clearly using this method. The method in this paper can be used in fuzzy process monitoring and control.

Keywords Petri nets, Linear logic, Fuzzy time Petri Nets, Temporal reasoning

Petri 网是一种用网状图形表示系统模型的方法,它能够从组织结构、控制和管理的角度,精确描述系统中事件(变迁)之间的依赖(顺序)和不依赖(并发)关系。但传统的 Petri 网理论其不足之处在于:它的分析方法主要是可达树分析法和线性代数描述法。可达树分析法是针对某一个初始标识的,一个新的初始标识就意味着需要重新构造可达状态图;当系统存在较多并发冲突的变迁时,会出现状态空间爆炸问题;不能区分变迁之间的并发和冲突关系;而线性代数描述法没有考虑触发顺序和关联矩阵中可能的闭环消元情况。

线性逻辑是由 J. Y. Girard^[1]在经典逻辑的基础上提出的,但对经典逻辑做了很大的变动,它表示了状态的概念,命题在推理过程中表示资源,当命题作为假设而产生结论时,这些作为假设的命题则失去真值,状态的变化由假设和结论的因果关系表示,与 Petri 网中状态转移的概念相一致性。目前一些文章^[2]中提到用线性逻辑对 Petri 网分析的思想,但未能全面地描述 Petri 网中的并发和冲突等问题,并不能很好地体现线性逻辑推理的优越性。

在实时系统中,存在时间约束问题,这就需要有一种时间推理方法对系统进行分析,传统的方法^[3]主要基于可达图,都是建立一个从初始标识到各可达标识的全部路径。在实际系统中,经常存在不确定的因素,例如不确定的事件发生时间和不确定的过程进行时间,这就需要处理模糊时间问题,一些文献^[3,4]提出了用模糊时间 Petri 网表达不确定时间和时间区间问题,文[5]证明了用 Petri 网比状态转移图能够更好地建模动态系统,但未能提出有效的推理方法,文[6]提出用模糊可达图进行模糊时间推理,它本质上基于可达图

分析,故存在可达图分析法的不足,现存推理方法主要是不能很好地描述和区分变迁的并发和冲突关系。

本文提出用线性逻辑描述 Petri 网结构,定义了 Petri 网中变迁之间的各种触发关系和规则,并在此基础上全面而准确地证明 Petri 网的推理运行过程,以提高对 Petri 网的描述和分析能力,并在一种模糊时间 Petri 网模型的基础上,提出了基于线性逻辑的时间推理方法,通过一个例子证明了其应用。

1 用线性逻辑描述 Petri 网

1.1 线性逻辑中的主要符号

线性逻辑中与 Petri 网描述有关的主要连接符有:

- ① 表示资源之间“与”的关系,与经典逻辑中的“and”具有基本相同的性质,但它不象“and”具有单义性,例如,对于命题 A , A and $A=A$,而 $A \otimes A = \Delta A$;
- ② 表示资源间“或”的关系,与经典逻辑中的“or”具有相同的性质,例如 $A \oplus A = A$;
- ③ 表示资源之间的因果关系。例如 $A \multimap B$ 表示由 A 命题产生 B 命题,而 A 命题不再成立,不象经典逻辑中, $A \Rightarrow B$ 表示 A 命题成立可导致 B 命题成立,且 A 命题仍然成立。
- ④ 表示资源之间的可证明关系。例如 $\Gamma \multimap \Delta$ 表示 Δ 可以由 Γ 证明得出,其中 Γ, Δ 为资源集合。

1.2 用线性逻辑描述 Petri 网的结构

任何 Petri 网都可以用线性逻辑算式描述。首先,可以用线性逻辑表示 Petri 网中的 token 分布,即表示 Petri 网的标识。对于任一 Petri 网,其当前标识可表示为:

$$M = \sum_{p \in P} n_p P_p$$

n_p is the number of marked places

高梅梅 博士生,研究方向:离散事件系统理论及应用,人工智能。吴智铭 教授,博士生导师,研究领域:离散事件动态系统和混杂系统,CIMS。

其中 P_i 为库所名称, m_i 表示库所 P_i 中有 m_i 个 token。这样表示 Petri 网的标识只用列出有 token 的库所, 不用列出所有的库所, 如 $A \leq B$ 则表示库所 A 和 B 中各有一个 token。

另外, Petri 网中的变迁表示了输入库所和输出库所之间的因果关系, 且变迁的触发意味着 token 从输入库所到输出库所的转移, 所以可用线性逻辑中的符号“ $\multimap$ ”描述 Petri 网中的变迁, “ $\multimap$ ”的左边表示是变迁触发所需要的最小标识, 右边表示由最小标识经变迁触发后所可达的标识。即对于 Petri 网中的每个变迁 t_i :

$$t_i = pre(t_i) \multimap post(t_i)$$

例如图 1 中的变迁 t_1 的 $pre(t_1) = A \leq A$, $post(t_1) = B \leq C$, 则 t_1 可定义为:

$$t_1: 2A \multimap B \leq C$$

同理 t_2 可定义为:

$$t_2: B \leq D \multimap C$$

这样通过符号“ $\leq$ ”和“ $\multimap$ ”定义 Petri 网中的变迁, 实际上就对 Petri 网的结构进行了描述。例如图 1 中 Petri 网的结构可描述如下:

$$t_1: 2A \multimap B \leq D$$

$$t_2: B \leq D \multimap C$$

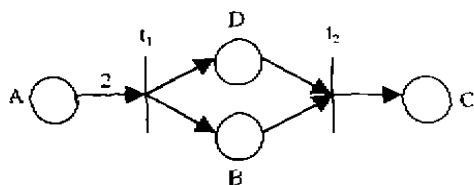


图 1 一个简单的 Petri 网模型

2 变迁之间不同的触发规则

2.1 规则的定义

为了表示 Petri 网中变迁之间的不同触发关系, 我们在线性逻辑的基础上, 定义了三种规则 (SEQ 规则、PAR 规则和 ALT 规则) 分别表示变迁触发的顺序、并发和冲突关系。

定义 1 (SEQ 规则: 变迁顺序触发) 对于两个变迁 $t_1: M \multimap K$ 和 $t_2: K \multimap L$, 若 t_2 的触发以 t_1 的触发为条件, 则 t_1 和 t_2 是顺序触发关系。SEQ 规则表示为:

$$\frac{t_1: M \multimap K \quad t_2: K \multimap N}{t_1 \cdot t_2: M \multimap N} \text{SEQ}$$

其中连接符“ $\cdot$ ”表示变迁顺序触发。

定义 2 (PAR 规则) 变迁并发或互不相关独立触发。对于两个变迁 $t_1: M \multimap K$ 和 $t_2: L \multimap N$, PAR 规则表示为:

$$\frac{t_1: M \multimap K \quad t_2: L \multimap N}{t_1 \parallel t_2: M \leq L \multimap K \leq N} \text{PAR}$$

其中连接符“ $\parallel$ ”表示变迁独立触发。

• 12 •

定义 3 (ALT 规则: 变迁并发冲突) 对于两个变迁 $t_1: M \multimap K$ 和 $t_2: M \multimap L$, 若 $pre(t_1)$ 和 $pre(t_2)$ 中含有相同的库所, 则 t_1 和 t_2 存在并发冲突。对于有限的输入标识 M , t_1 和 t_2 只能选择其一触发。ALT 规则表示为:

$$\frac{t_1: M \multimap K \quad t_2: M \multimap L}{t_1 \sqcup t_2: M \multimap K \sqcup L} \text{ALT}$$

其中连接符“ $\sqcup$ ”表示变迁并发冲突。

以上讨论了用线性逻辑对变迁或变迁序列的定义, 它是基于网结构的, 与初始标识无关。但实际上, 由于初始标识的不同, 变迁之间的触发关系不是绝对的, 而是可变的。例如在上例中, 若初始标识中库所 A, B 和 D 中都有 token (A 中有两个 token), 则 t_1 和 t_2 可以并发。在这种情况下, t_1 和 t_2 的触发序列的定义为:

$$\frac{t_1: 2A \multimap B \leq D \quad t_2: B \leq D \multimap C}{t_1 \parallel t_2: 2A \leq B \leq D \multimap B \leq D \leq C} \text{PAR}$$

上式表示 t_1 和 t_2 并发所需的最小标识为 $2A \leq B \leq D$ 。

2.2 用线性逻辑描述 Petri 网的运行行为

根据 Petri 网的描述, 线性逻辑可以通过描述标识的变化来表示 Petri 网运行的行为, 在 M 标识下, 经变迁 t_i 触发后, Petri 网的运行行为可表示为:

$$M, t_i \vdash M'$$

其中 M' 为 Petri 网运行后的标识, 上式表示系统从状态 M 到状态 M' 是可证明的。同理经变迁序列 s 触发, Petri 网的运行行为可表示为:

$$M, s \vdash M'$$

这里值得再一次说明的是符号“ $\vdash$ ”和“ $\vdash$ ”的不同, 在描述 Petri 网的结构时, 我们用符号“ $\multimap$ ”定义 Petri 网的变迁和变迁序列, “ $\multimap$ ”的左边是变迁触发所需的最小标识, 即 $pre(t)$; 而“ $\vdash$ ”表示 Petri 网的运行, 它允许描述系统在某初始标识下 (该标识大于等于变迁触发的最小标识) 的运行状况。

基于上节所定义的变迁触发规则, 我们可以对系统的触发序列进行组合, 即:

$$\frac{M_1, s_1 \vdash M_2 \quad M_2, s_2 \vdash M_3}{M_1, s_1 \cdot s_2 \vdash M_3} \text{SEQ}$$

$$\frac{M_1, s_1 \vdash M_2 \quad M_1, s_2 \vdash M_3}{M_1 \leq M_2, s_1 \parallel s_2 \vdash M_2 \leq M_3} \text{PAR}$$

$$\frac{M_1, s_1 \vdash M_2 \quad M_1, s_2 \vdash M_3}{M_1, s_1 \sqcup s_2 \vdash M_2 \sqcup M_3} \text{ALT}$$

由此, 我们就可以采用多种规则的组合从而描述和证明 Petri 网的行为。

3 模糊时间 Petri 网

为了定义模糊时间 Petri 网, 我们先定义一模糊时间区间 $\pi(\tau)$ 。

定义 4 一模糊时间区间 $\pi(\tau) = [\tau^-, \tau^+, \tau^-, \tau^+]$,

$\tau \leq \tau^- \leq \tau^+ \leq \tau$, $\tau^-, \tau^+, \tau \in \mathbb{Q}^+$, 其中 $\mathbb{Q}^+$ 为非负实数集, 表示“时间集合”:

$$\pi(\tau) = \{[\tau^-, \tau^+, \tau^-, \tau] | \tau^- \in \mathbb{Q}^+ \wedge \tau^+, \tau \in \mathbb{Q}^+ \cup \{\infty\} \wedge \forall \tau \in [0, \tau^-], \pi(\tau) = 0 \wedge \forall \tau \in [\tau^-, \tau^+], \pi(\tau) = (\tau - \tau^-) / (\tau^+ - \tau^-) \wedge \forall \tau \in [\tau^+, \tau], \pi(\tau) = 1 \wedge \forall \tau \in [\tau, \infty], \pi(\tau) = 0\}$$

我们把这样的时间区间集合记为 Π .

定义 5 模糊时间 Petri 网可以定义为六元组, $\text{FTPN} = \langle P, T, \text{Pre}, \text{Post}, M_0, \theta \rangle$, 其中 P 为所有库所的集合 $P \neq \emptyset$; T 为所有变迁的集合, $T \neq \emptyset$; $\text{Pre}: P \rightarrow T$; Post 为后向关联函数, $\text{Post}: T \rightarrow P$; M_0 为 P 上的初始标识; $\theta: T \rightarrow \Pi$, 为一映射, 称为模糊静态时间区间。

在模糊时间 Petri 网中, 每个变迁被赋予一模糊时间区间 $\theta(t) = [\tau^-, \tau^+, \tau^-, \tau]$, τ^-, τ^+, τ^- 为非负实数, $\tau^- \leq \tau^- \leq \tau^+ \leq \tau$, 模糊静态时间区间呈梯形分布 (图 2)。时间 τ^-, τ^+, τ^- 是相对时间, 它们相对于变迁使能的时刻, 它们表示变迁从使能时刻到触发时刻的延迟。假如变迁在 τ 时刻使能, 那么变迁 t 不能在 $\tau + \tau^-$ (记为 d_t^-) 之前或 $\tau + \tau^+$ (记为 d_t^+) 时刻之后触发, 否则说明系统出现故障。如果变迁 t 在 $\tau + \tau^-$ (记为 d_t^-) 和 $\tau + \tau^+$ (记为 d_t^+) 时刻之间触发, 表示系统工作在正常状态下, 我们把区间 $[\tau^-, \tau^+]$ 称为“核心区间”; 如果变迁 t 在 $\tau + \tau^-$ 和 $\tau + \tau^+$ 时刻之间或 $\tau + \tau^+$ 和 $\tau + \tau^-$ 时刻之间触发, 表示系统工作在非正常状态下, 但也是可能的状态, 我们把整个区间 $[\tau^-, \tau^-]$ 称为“允许区间”(support interval)。

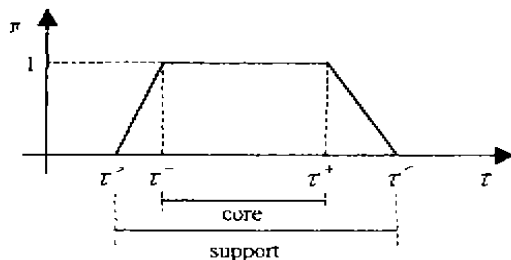


图 2 模糊时间区间

4 模糊时间 Petri 网的时间推理

前面论述了用线性逻辑进行 Petri 网结构的表示和运行行为的证明方法, 为了基于模糊时间 Petri 网, 利用线性逻辑进行时间推理, 对应于三种变迁触发关系, 我们定义三种时间运算关系。

定义 6 对于两个模糊时间区间 $l, l', l = [l^-, l^+, l^-, l^+]$, $l' = [l'^-, l'^+, l'^-, l'^+]$, “加和时间”(additive

date) $l \oplus l' = \text{ad}(l, l')$, 则:

$$l_+ = l_+ + l'_+$$

$$l_- = l_- + l'_-$$

$$l^+ = l^+ + l'^+$$

$$l^- = l^- + l'^-$$

对于顺序触发的变迁(我们定义的 SEQ 规则), 必须在每个变迁依次触发结束, 它们所组成的变迁序列触发才能完成, 所以顺序触发的变迁组成的变迁序列以“加和时间”触发。

定义 7 对于两个模糊时间区间 $l, l', l = [l^-, l^+, l^-, l^+]$, $l' = [l'^-, l'^+, l'^-, l'^+]$, “较晚时间”(later date) $l \ominus l' = \text{ld}(l, l')$, 则:

$$l_+ = \max(l_+, l'_+)$$

$$l_- = \max(l_-, l'_-)$$

$$l^+ = \max(l^+, l'^+)$$

$$l^- = \max(l^-, l'^-)$$

对于并发关系的变迁(我们定义的 PAR 规则), 它们所组成的变迁序列触发完成时间是最晚触发的变迁触发完毕的时间, 所以并发的变迁组成的变迁序列以“较晚时间”触发。

定义 8 对于两个模糊时间区间 $l, l', l = [l^-, l^+, l^-, l^+]$, $l' = [l'^-, l'^+, l'^-, l'^+]$, “较宽时间”(wider date) $l \oplus l' = \text{wd}(l, l')$, 则:

$$l_+ = \min(l_+, l'_+)$$

$$l_- = \min(l_-, l'_-)$$

$$l^+ = \max(l^+, l'^+)$$

$$l^- = \max(l^-, l'^-)$$

对于触发冲突的变迁(我们定义的 ALT 规则), 它们触发完成时间是由各变迁中最小的最早触发时间和最大的最晚触发时间决定的, 所以触发冲突的变迁组成的变迁序列可能在最宽的时间区间内完成触发。

另外, 必须注意, 在 Petri 网结构上处于冲突关系的变迁, 而在模糊时间 Petri 网实际运行中, 由于时间的约束不一定存在冲突, 这一点我们有定理如下。

定理 1 在 Petri 网结构上处于同一冲突下的所有变迁 $t_1, t_2, \dots, t_n$ 中, $\theta(t_i) = [\tau_i^-, \tau_i^+, \tau_i^-, \tau_i^+]$, $i = 1, \dots, n$, 若存在某变迁 $t_k, \tau_i^+ < \tau_k^-, 1 \leq i \leq n, k = 1, \dots, n, k \neq j$, 则在模糊时间 Petri 网的实际运行中变迁 $t_1, t_2, \dots, t_n$ 之间不存在冲突, 变迁 $t_1, t_2, \dots, t_n$ 使能后, 系统将在模糊时间区间 $[\tau_k^-, \tau_k^+, \tau_k^-, \tau_k^+]$ 内沿 t_k 运行。

证明: 对于结构上处于同一冲突下的变迁, 若其中存在某变迁可能的最迟触发时间小于所有其他变迁可能的最早触发时间, 此变迁必将在其可能的最迟触发时间之前触发, 所以定理 1 显然成立。

推论 1 在 Petri 网结构上处于同一冲突下的所

有变迁 $t_1, t_2, \dots, t_n$ 中, $\theta(t_i) = [\tau_i^-, \tau_i^+, \tau_i^-, \tau_i^+]$, $i = 1, \dots, n$. 若对于某变迁 t_k , 存在另一变迁 t_i , $\tau_i^- < \tau_k^-$, $1 \leq i, k \leq n, k \neq i$, 则 t_k 将在竞争中被淘汰。

因此, 在分析模糊时间 Petri 网的运行行为时, 对于结构上处于同一冲突下的变迁, 应分清它们之间的运行关系。

定理 2 非模糊时间 Petri 网的定义和时间推理可用模糊时间 Petri 网表示。

在非模糊时间 Petri 网模型中, 变迁上的时间区间只由两个时刻决定, 即最早触发时间和最晚触发时间, 变迁早于最早触发时间和晚于最晚触发时间触发都认为是不可可能的, 即在模糊时间 Petri 网的定义中, 若 $\tau^- = \tau^+$ 且 $\tau^+ = \tau^-$, 则其就成为非模糊时间 Petri 网。时间推理方法也是一样, 只是 $\tau^- = \tau^+$, $\tau^+ = \tau^-$ 。

5 举例

我们以工业生产中一段过程为例, 说明本文的方法。加工设备加工完一个工件 (t_1) 后:

• AGV 来取工件 (p_2), AGV 将加工好的工件运到缓冲区 (t_2), AGV 在缓冲区处放下已加工好的工件并取走待加工的工件 (t_3), AGV 把待加工的工件运到机床 (t_4), 待加工工件在机床上安装完毕, (t_5), 待加工工件等待加工 (p_7);

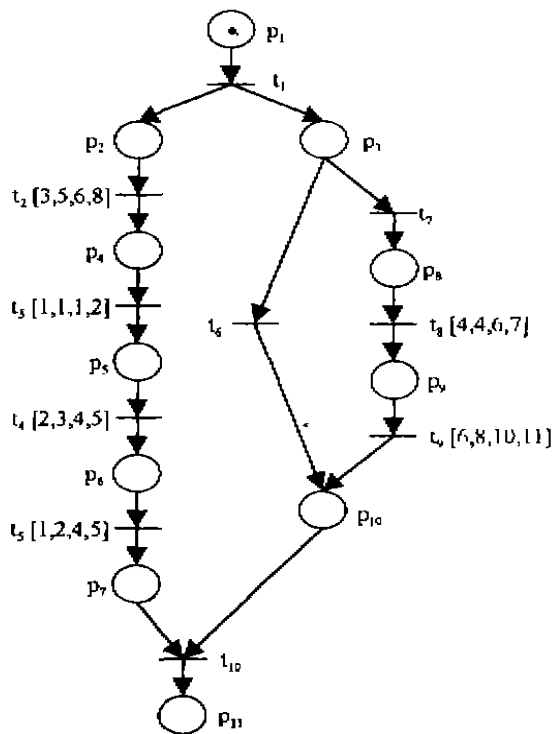


图 3 所举例子的模糊时间 Petri 网模型

• 若两个加工过程所需刀具一样, 机床等待加工下一工件 (t_{10});

• 若两个加工过程所需刀具不一样, 机械手来取刀具 (t_1), 机械手取走已用刀具并取来所需刀具 (t_2), 安装刀具完成 (t_3), 机床等待加工工件 (p_{10});

待加工工件和刀具都准备好后, 机床开始加工待加工工件 (t_{10})。

我们根据模糊时间 Petri 网的定义, 可得所举例子的模糊时间 Petri 网模型如图 3 所示, 其中没有标注时间区间的变迁表示该变迁使能后可立即触发, 即时间区间为 $[0, 0, 0, 0]$ 。我们认为系统的初始时刻为 0 时刻, 系统的初始标识 $M_0 = p_1$ (篇幅所限, 此 Petri 网结构的线性逻辑描述略, 可参见第 1.2 节的定义)。

我们可以用本文的方法对此加工过程推理:

$$\frac{t_2: p_2 \multimap p_4 \quad t_3: p_4 \multimap p_5 \quad t_4: p_5 \multimap p_6 \quad t_5: p_6 \multimap p_7}{t_2 * t_3 * t_4 * t_5: p_2 \multimap p_7} \text{SEQ}$$

$$\text{同理可得: } t_1 * t_3 * t_4 * t_5: p_1 \multimap p_{10}$$

$$\frac{t_6: p_1 \multimap p_{10} \quad t_7: t_6 * t_8 * t_9: p_1 \multimap p_{10}}{t_6 / (t_7 * t_8 * t_9): p_1 \multimap p_{10}} \text{ALT}$$

为了简化表示, 令 $s_1 = t_2 * t_3 * t_4 * t_5, s_2 = t_6 / (t_7 * t_8 * t_9)$, 即

$$s_1: p_2 \multimap p_7$$

$$s_2: p_1 \multimap p_{10}$$

$$\frac{s_1: p_2 \multimap p_7 \quad s_2: p_1 \multimap p_{10}}{s_1 \parallel s_2: p_2 \multimap p_7 \quad p_1 \multimap p_{10}} \text{PAR}$$

$$\frac{t_1: p_1 \multimap p_2 \quad s_1 \parallel s_2: p_2 \multimap p_7 \quad p_1 \multimap p_{10}}{t_1 * (s_1 \parallel s_2): p_1 \multimap p_7 \quad p_1 \multimap p_{10}} \text{SEQ}$$

$$\frac{t_1 * (s_1 \parallel s_2): p_1 \multimap p_7 \quad p_1 \multimap p_{10} \quad t_{10}: p_7 \multimap p_{10} \quad p_1 \multimap p_{10}}{t_1 * (s_1 \parallel s_2) * t_{10}: p_1 \multimap p_{10}} \text{SEQ}$$

令 $s_3 = s_1 \parallel s_2, s_4 = t_{10} * (s_1 \parallel s_2) * t_{10}$, 则在初始标识 $M_0 = p_1$ 下, 系统的运行行为为:

$$p_1, s_4 \vdash p_{11}$$

下面我们根据本文的定义对系统进行时间推理:

$$\theta(s_1) = \text{ad}(\theta(t_2), \theta(t_3), \theta(t_4), \theta(t_5))$$

$$= [7, 11, 15, 20]$$

$$\theta(s_2) = \text{wd}(\theta(t_6), \text{ad}(\theta(t_7), \theta(t_8), \theta(t_9)))$$

$$= [0, 0, 16, 18]$$

$$\theta(s_3) = \text{ld}(\theta(s_1), \theta(s_2)) = [7, 11, 16, 20]$$

$$\theta(s_4) = \text{ad}(\theta(t_1), \theta(s_3), \theta(t_{10})) = [7, 11, 16, 20]$$

可见, 整个 Petri 网的运行时间为 $[7, 11, 16, 20]$, 如果另一机床加工完一个工件到开始加工下个工件需要延迟的时间为 t , 则

• 当 $11 \leq t \leq 16$ 时, 系统工作正常;

• 当 $7 \leq t \leq 11$ 或 $16 \leq t \leq 20$, 系统非正常工作, 但未出现致命故障;

• 当 $t \leq 7$ 或 $t \geq 20$, 系统出现故障。

(下转第 31 页)

树且该树为组内所有成员共享,收发分组通过同一棵树,因而开销较小。总体而言,该方法的网络资源开销小于捎带方法而大于目录查询和显式管理库配置。

9)健壮性 目录查询和显式管理库配置以目录服务器和 MIB 管理站为核心,一旦它们出现问题,就无

法提供成员信息。因此从可靠性出发,应设置备份服务器。捎带方法和多播方法不依赖于某一网络组件,因而健壮性很好。

表 1 总结了上述情况。

表 1 各种方法的比较

特性 \ 方法	目录查询	显式管理库配置	路由协议捎带方法	多播方法
动态性	不好 * 需数据库同步机制	不好 * 需数据库同步机制	好	好
拓扑适应性	好	好	支持全连接结构;支持任意拓扑结构比较困难	支持全连接结构;支持任意拓扑结构比较困难
跨区域性	较好 * 需目录间协议	不好	好	好
安全性	有	有	不足	有
对现有路由协议的影响	无	无	有 * 需修改路由协议选项	无
效率	高	高	低	低
基础设施	需目录服务器	需 MIB 管理站	路径上所有路由识别修改后协议	VPRN 成员路由器支持多播
网络资源开销	小	小	大	较大
健壮性	较差	较差	好	好

结束语 本文总结了 VPRN 成员信息传播的几种方法,并对其在动态性、拓扑适应性、跨区域性、安全性、对现有路由协议的影响、效率、基础设施、网络资源开销和健壮性等方面进行了比较。每种方法各有优点,也有不尽人意之处,只有将多种方法结合起来才能高效、可靠、简洁地进行 VPRN 成员信息传播。

参考文献

- 1 Ferguson P, Huston G. What is a VPN?. Available at: URL: <http://www.employees.org:80/~ferguson/vpn.pdf>, April 1998

- 2 Gleeson, et al. A Framework for IP based Virtual Private Networks. draft-gleeson--vpn-framework-01.txt, Feb. 1999
- 3 Muthukrishnan and Mahis. Core IP VPN Architecture. draft-muthukrishnan-corevpn-arch-00.txt, Oct 1998
- 4 Fox B, gleeson B. Virtual Private Networks Identifier. draft-ietf-100-vpn-id-02.txt, July 1999
- 5 Jameson, et al. MPLS VPN Architecture. draft-jameson-mpls-vpn-00.txt, Aug. 1998
- 6 Chandra R, Frana P. BGP Communities Attribute. RFC 1998
- 7 Coltun R. The OSPF Opaque LSA Option. RFC 2370
- 8 Waller D, et al. Key Management for Multicast: Issues and Architecture. draft-wallner-key-arch-01.txt
- 9 Gupta V. Secure, Remote Access over the Internet using IPSec. draft-gupta-tpsec-remote-access-02.txt, Jun 1999

(上接第 14 页)

参考文献

- 1 Girard J Y. Linear Logic. Theoretical Computer Science, n°150, 1987
- 2 Valette R, Courvoisier M. Petri nets and Artificial Intelligence. Modern Tools for Manufacturing Systems, Elsevier Science Publishers, 1993. 385~405
- 3 Berthomieu B, Diaz M. Modeling and verification of time dependent systems using Petri nets. IEEE Trans. on Soft-

ware Eng., 1991, 17(3): 259~273

- 4 Yao Y. A Petri net model for temporal representations and reasoning. IEEE Trans. on Systems, Man and Cybernetics, 1994, 24(9): 1374~1382
- 5 Cardoso J, et al. Petri net based reasoning for the diagnosis of dynamic discrete event systems. In: 6th Intl. Fuzzy Systems Association World Congress, IFSA'95, Sao Paulo, Brazil, July, 1995. Vol. 1. 333~336
- 6 de Figueiredo J C A, Perkins A. Fault and Timing Analysis in Real-Time Distributed Systems: A Fuzzy Time Petri-Net-Based Approach. Intl. J. Fuzzy Sets and Systems, 1996, 83(3): 143~168