

嵌入式通信中间件的马尔可夫路由决策与选择

贾智平¹, 刘甜甜¹, 张承慧², Edwin H.-M. Sha³

(1. 山东大学计算机科学与技术学院, 山东济南 250061; 2. 山东大学控制科学与工程学院, 山东济南 250061;

3. 美国德州大学达拉斯分校计算机科学系, 理查森, 德克萨斯 75083)

摘 要: 针对嵌入式设备互连环境和嵌入式通信中间件的特点, 结合马尔可夫决策过程理论, 建立了解决路由问题的有限阶段模型, 并修改马尔可夫有限阶段模型的向后递归迭代算法, 提出了马尔可夫有限阶段决策路由算法 MFHDR (Markov Finite Horizon Decision Routing). 该算法具有分布计算和自我学习的特性, 从而降低了单台嵌入式设备的工作强度, 均衡了各台设备的负载, 具有较好的时间和空间复杂度, 并且能够有效的避免环路产生.

关键词: 嵌入式设备互连; 嵌入式通信中间件; 马尔可夫决策过程; 有限阶段模型; 分布计算

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2007) 07-1228-06

Markov Route Decision in Embedded Communication Middleware

JIA Zhir ping¹, LIU Tian tian¹, ZHANG Cheng hui², Edwin H.-M. Sha³

(1. School of Computer Science and Technology, Shandong University, Jinan, Shandong 250061, China;

2. School of Control Science and Engineering, Shandong University, Jinan, Shandong 250061, China;

3. Department of Computer Science, University of Texas at Dallas, Richardson, Texas 75083, USA)

Abstract: Combining with the theory of Markov decision process, a finite horizon model to resolve the routing problem is established in view of the features of embedded equipment interconnection environment and embedded communication middleware. The backward recursion iterative algorithm of Markov finite horizon model is modified, and Markov Finite Horizon Decision Routing (MFHDR) algorithm is then proposed. MFHDR algorithm is characteristic by distributed computing and self studying, thus reduces work intensity of single embedded equipment, balances workload between each equipment, and has a better time and space complexity and is loop free.

Key words: interconnection of embedded equipment; embedded communication middleware; Markov decision process; finite horizon model; distributed computing

1 引言

嵌入式系统的普遍使用导致它们的互连成为一种巨大的应用需求. 但嵌入式设备种类繁多, 系统和网络各不相同, 如何屏蔽这种异构性, 使之可以以统一的方式使用网络协议接口实现信息交互已成为目前研究热点之一. 通信中间件便是解决上述问题的一种方法, 它能把应用扩展到不同的操作系统和不同的网络环境中, 在嵌入式设备之间建立逻辑通道, 屏蔽不同设备系统和网络协议的异构性, 完成从发送方到接收方的通信, 从而发送控制命令和各种信息以达到交互的目的.

由于嵌入式本身资源有限, 对实时性的要求比较高, 要开发嵌入式设备之间的通信中间件, 就比在通用

主机之间要受到更多限制, 不能采用常见的局域网, 广域网等已成熟的网络技术. 设备间没有路由器来转发消息, 因此, 嵌入式通信中间件应位于链路层之上. 各个设备是对等关系, 既要完成自己本身的功能, 又要在消息通信中起到路由和中转信息的作用, 这一点上有点类似于 AdHoc 网络的多跳转发^[1]. 因此这类通信中间件应占用较少的资源, 花费较短的时间, 从而不增加嵌入式设备本身的功能负担. 而通信的计算量有很大一部分来自于从发送方到接收方的路径选择上. 因此, 研发一种低时间和空间复杂度的路由算法便是解决问题的关键. 考虑到嵌入式网络环境中多点通信 (多源多目或单源多目) 的频率和概率并不大, 本文主要探讨单源单目的按需路径选择问题.

路由, 从整体上看是选择从源到目的一条路径, 对这个问题的研究非常多^[2-6], 但这些研究和改进技术都是针对普通的网络, 不能直接用于嵌入式互连网络. 对于嵌入式环境下的路由, 具体到每一个嵌入式设备(充当路由器功能)其实是一个决策问题, 决定从它的外出线路中选哪一条来传输当前消息. 嵌入式设备存储能力有限, 所以不便保存网络的全局拓扑信息. 可以采取的解决办法是: 每个设备在照顾全局的前提下进行本地决策^[7], 产生一个决策序列从而确定一条路径. 研究中发现, 这个路由过程符合马尔可夫决策过程. 而且, 使用这种本地决策的分布式算法可以将路由工作分布到各个设备之上, 有利于平衡各个设备的工作负载, 降低单个设备的工作强度, 同时各个设备具有了自我学习的能力. 因此, 本文应用马尔可夫决策理论来求解这个路径选择问题.

2 马尔可夫决策过程

马尔可夫过程是一类随机序列决策的过程^[8,9]. 在各个时刻根据观察到的状态, 从它的允许行动集中选用一个, 称为“决策”, 从而决定了系统下次状态的转移和相应的运行效果——报酬, 这样周期地或连续地进行下去, 从而确定一个决策序列, 称之为“策略”.

马尔可夫过程的特点是无后效性, 用五元组 $\{T, S, A(i), p(\cdot|i, a), r(i, a)\}$ 表示, 其中: $T = \{0, 1, \dots\}$ 是决策时刻集; S 是所有可能状态集, i 是一个系统状态, $i \in S$; $A(i)$ 是行动集, 表示系统在状态 i 时的可用行动集, a 是一个行动, $a \in A(i)$; $p(\cdot|i, a)$ 给出了转移概率规律, $p(j|i, a)$ 表示当前状态为 i , 选择行动 a , 到下一个决策时刻转移到状态 j 的概率; $r(i, a)$ 是实值报酬函数. $A(i), p(j|i, a), r(i, a)$ 只依赖于当前的系统状态和选取的行动, 与过去的历史无关. 这个模型看似过于受限, 但它其实已涵盖大部分的序列决策模型. 而要解决的路径选择问题也正是符合这一模型特点的.

策略的全体称为策略空间 Π 它包括所有与历史有关的或无关的, 确定性的或随机的, 平稳的等多种策略类, 在此只介绍马氏决策的策略类: 马氏策略类.

定义 1 决策函数 f : 对于每个 $i \in S$, 有 $f(i) \in A(i)$, 则称为确定性的决策规则或决策函数. f 的全体记做 F .

定义 2 马氏策略 π : 一个决策函数序列 $\pi = (f_0, f_1, f_2, \dots), f_t \in F, t \in \mathbb{N}$. 其中 f_t 是决策时刻 t 的决策函数, 不依赖时刻 t 以前系统的历史, $t \in \mathbb{N}$. 马氏策略的全体集合记做 Π_m , 称为马氏策略类.

对应于一个马氏策略可以产生一个报酬序列 $(R_0(\pi), R_1(\pi), \dots)$, 其中 $R_t(\pi) = r(i_t, a_t), t \in \mathbb{N}$. 而序列决策问题就是要在第一个决策时刻之前就预先选

好策略, 使得报酬序列的某个效用函数值——准则在这个策略下达到最优.

3 路由的决策模型

3.1 路由的有限阶段模型

路径选择的过程也是一类序列决策过程. 将嵌入式设备的互连拓扑抽象成有向图, 每一个设备就是图中的节点. 消息传到一个节点, 进行路由就是要在它的出边上选择一条来转发此消息. 节点选择链路时也正满足马氏过程的无后效性.

不同于通用网络的是: 嵌入式环境下的决策要同时考虑到其出边及其相连的邻居节点 v 的多个因素, 例如邻居节点的处理能力, 内存资源情况, 能量供应, 连通度, 物理移动性, 链路的延时, 传送速度和流量负载等. 这些因素构成费用向量 CV 通过相邻节点互相交换进行更新, 每个节点使用这些信息形成一个决策的度量费用来实施决策.

路由的跳数虽然是不可预知的, 但网络传送数据往往要避免过多的跳步, 那么选择时刻便可以看作是有限的(ttl). 因此可以建立路由的有限阶段的马氏决策模型.

定义 3 有限阶段的马氏决策模型^[9]: 指离散时间决策时刻的马氏决策模型, 其阶段是有限的. 该模型使用马氏决策五元组 $\{T, S, A(i), p(\cdot|i, a), r(i, a)\}$ 定义, 其中 $T = \{0, 1, \dots, N-1\}, 0 \leq N < \infty, r(i, a)$ 是有界报酬函数. 在选定一个策略后, 决策者在阶段 $0, 1, \dots, N$ 时依一定的概率收到一串报酬, 将其累加就是该模型的具体效用函数.

设有向图 $G(V, E)$, 如图 1, $|V|, |E|$ 分别表示节点和边的数目. 已知源节点为 s , 目的结点为 d , 方便起见对节点使用数字编号表示, 从 0 开始编号, 则 $s, d \in [0, |V|-1]$.

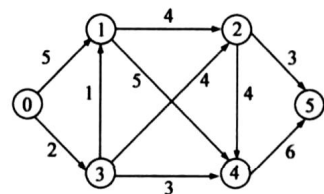


图 1 有向图 G

路由的决策模型建立如下:

决策时刻: $T = \{0, 1, \dots, ttl\}$, 其中 ttl 表示跳数的最大允许值, 在此有 $ttl = |V| - 1$, 即最多允许跳过除去源节点的所有的节点;

可能的状态: $S = V = [0, 1, \dots, |V|-1]$, 即一个节点表示一个状态;

可能的行动集: $A(i) = e_{ij}, e_{ij} \in E$ 即一个状态的行动集是它代表的节点的所有出边;

报酬值: $r_t(i, a) = r_t(i, e_{ij}) = c(e_{ij})$, $i, j \in S$, $e_{ij} \in E$, $t \in [0, ul]$, 其中 $c_t(e_{ij})$ 是链路的度量费用. 定义公式见后文. $r_N(i) = 0$, $i \in S$, $N \in [0, ul]$, N 是结束时刻.

转移概率:

$$p_t(j' | i, e_{ij}) = \begin{cases} 1, j' = j \\ 0, \text{其他} \end{cases} \quad i, j, j' \in S, e_{ij} \in E, t \in [0, ul].$$

这样就建立好了路径选择的有限阶段的马氏决策问题.

在嵌入式环境中, 下一跳的选择是受很多因素共同影响的, 这不同于一般的网络环境往往只考虑链路相关的带宽和延时. 在此将各个因素整合到一起, 定义费用公式如下:

$$c(e_{ij}) = \frac{1}{\alpha \cdot cd + \beta \cdot me + \gamma \cdot qp + \lambda \cdot lp} + \frac{\eta \cdot lc}{lc - lu} + \frac{1}{\varphi \cdot f} + \mu \cdot mo + \omega \cdot dl \quad (1)$$

其中 cd 代表 e_{ij} 边的另一个节点 j 当前的处理器处理能力, 有 $0 < cd < 1$, 对应于节点 j 当前处理器空闲百分比; me 代表节点 j 的内存资源剩余容量, $0 < me < 1$, 对应于当前内存耗用剩余百分比; qp 代表节点 j 的能量供应, $0 < qp < 1$, 对应于当前能量的剩余百分比; lp 代表节点 j 的连通度, $0 < lp < 1$, 用公式 $p = lo / |E|$ 计算, 其中 lo 是该节点的出边数目; lc 代表边 e_{ij} 的总带宽, lu 表示当前流量负载, 则实际带宽余量为 $lc - lu$ (Mbit/s); dl 代表 e_{ij} 边的延迟, $0 < dl < 1$ (s); f 为本地节点对不同链路的偏好度 (Local Preference); mo 代表节点 j 的物理移动性, $0 < mo < 1$, 对应它的移动概率. 其中 $\alpha, \beta, \gamma, \lambda, \eta, \varphi, \mu, \omega$ 依次是各个因素的权重因子, 针对不同的应用背景而不同. 例如: 如果嵌入式设备对能量要求严格并且敏感, 那么 γ 则是一个较大的值; 如果嵌入式网络对服务质量加以要求, 那么 η 则可以设置为一个较大的值; 如果节点偏好使用本条链路的串口通信, 则 φ 较大.

3.2 模型的最优方程求解

对于上述有限阶段模型, 文献[8]给出了它的理论求解方法.

定义从时刻 t 到时刻 N 的期望报酬总和 $u_t^\pi(ht)$ 为:

$$u_t^\pi(h_t) = E\pi\left\{\sum_{n=t}^{N-1} R_n(\pi) + R_N(i_N) \mid h_t, Y_t = i_j\right\} \quad (2)$$

其中 $h_t, i_t, \pi, R_n(\pi)$ 分别与第2节中给出的符号含义相同, $R_N(i_N) = r(i_N, a_N)$, E_π 是期望算子, h_t 是从时刻 0 到时刻 t 的一条轨迹 $h_t = (i_0, a_0, i_1, a_1, \dots, i_{t-1}, a_{t-1}, i_t)$, $t \geq 0$.

定义最优方程为:

$$u_t(h_t) = \sup_{a \in A(i_t)} \{r_t(i_t, a) + \sum_{j \in S} p_t(j \mid i_t, a) u_{t+1}(h_t, a, j)\} \quad (3)$$

对于 $t = N$, 加上边界条件:

$$u_N(h_N) = r_N(i_N) \quad (4)$$

其中 $h_N = (h_{N-1}, a_{N-1}, i_N)$. 当式(3)的 \sup 可以达到时, 使用符号 \max 或 \min .

在此要说明的是: 式(2), 式(3), 式(4)都是通用公式, 在策略空间 Π 中进行选择, 它们是 h_t 的函数.

文献[8]证明: 有限决策模型在最优方程(式(3), 式(4))有解时存在最优的马氏策略, 并可以通过最优方程求解出最优的马氏策略. 因此在后文中就在这个定理成立的基础上, 对公式进行部分修改, 建立路由有限阶段模型在马氏策略类 Π_m^t 中的最优策略求解算法.

4 MFHDR 路由算法

4.1 MFHDR 算法的提出与描述

在马氏理论中, 有限阶段模型的最优策略求解一般采用向后递归迭代算法^[9]. 它是计算特定开始状态和终止状态之间的最优决策序列, 对应于上文提出的特定模型, 即计算单个源节点和单个目的节点之间的一条最短路径.

首先针对特定的模型对最优方程进行推导化简:

$$\begin{aligned} u_t(h_t) &= u_t(i_t) \\ &= \sup_{a \in A(i_t)} \{r_t(i_t, a) + \sum_{j \in S} p_t(j \mid i_t, a) u_{t+1}(i_p, a, j)\} \\ &= \min_{e_{ij} \in A(i_t)} \{r_t(i_t, e_{ij}) + \sum_{j \in S} p_t(j \mid i_t, e_{ij}) u_{t+1}(j)\} \\ &= \min_{e_{ij} \in A(i_t)} \{r_t(i_t, e_{ij}) + p_t(j \mid i_t, e_{ij}) u_{t+1}(j)\} \\ &= \min_{e_{ij} \in A(i_t)} \{r_t(i_t, e_{ij}) + u_{t+1}(j)\} \end{aligned}$$

引入记号:

$$\arg \min_{x \in X} g(x) = \{x' \in X \mid g(x') \leq g(x), \forall x \in X\};$$

对决策函数 $f(i)$ 做修改: 令 $f(i) = \text{Another}(a(i))$,

其中 $\text{Another}(e_{ij}) = j$, 即边 e_{ij} 的入端点.

对基本的向后递归迭代算法进行修改, 得到适应本文的应用环境 MFHDR 算法如下:

步骤 1 令 $t = 0, i_t = d$,

$$\begin{aligned} \text{初始化: } u_k^*(n) &= \infty, k = 0, 1, \dots, ul, n = \\ &0, 1, \dots, |V| - 1, n \neq d, \\ u_0^*(d) &= 0 \end{aligned}$$

步骤 2 令 $t+1 \Rightarrow t$, 如果 $t = ul$, 进入步骤 5; 否则, 进入步骤 3.

步骤 3 对所有 $i_t \in S, i_t \neq d$, 如果 $\forall e_{ij} \in E, \exists u_{t-1}^*(j) \neq \infty$, 计算:

$$u_t^*(i_t) = \min_{e_{ij} \in A(i_t)} \{r_t(i_t, e_{ij}) + u_{t-1}^*(j)\}$$

记:

$$A_t^*(i_t) = \arg \min_{e_{ij} \in A(i_t)} \{r_t(i_t, e_{ij}) + u_{t-1}^*(j)\};$$

并任意取定 $f_{t+1}^*(i_t) \in A_t^*(i_t)$, 这样就定义了 t 时刻

的决策规则 f_t^* . 进入步骤 4.

步骤 4 如果此时存在 $i_t \neq s$ 有 $u_t^*(i_t) \neq \infty$, 返回步骤 2, 否则进入步骤 5.

步骤 5 对于所有 $i \in S, i \neq d$, 计算: $u_{opt}(i) = \min_{i \in [0, n]} \{u_t^*(i)\}$ 及其对应的 $f_{opt}(i)$.

如果 $u_{opt}(s) = \infty$, 则无法计算出 s 到 d 的路径, 即不存在 s 到 d 的任何通路;

否则, 对应的: $\pi(s) = (s, f_{opt}(s), f_{opt}(f_{opt}(s)), f_{opt}(f_{opt}(f_{opt}(s))), \dots, d)$ 即是 s 到 d 最短路径. 算法终止.

对上述算法的说明: t 是算法的控制变量, 既能控制算法的终止, 又能结束环路的重复计算, 同时指示了当前决策时刻某路径中边的条数, 在此, 称 t 为“阶段数”. $u_t^*(i_t)$ 是节点 i 到目的节点 d 的 t 阶段最短距离 (后简称“阶段距离”), $A_t^*(i_t)$ 中保存的是使之具有最短距离 $u_t^*(i_t)$ 的可选择行动. 如果 $A_t^*(i_t)$ 中的元素个数大于 1, 则表明 i 到 d 的 t 阶段最短路径不止一条, 这时可以任意选择一条的入端点作为当前的决策行动 $f_t^*(i_t)$. 步骤 4 也用来控制算法的流程, 当已无有效的 $u_t^*(i_t)$ 可用时, 可以及时的跳出由 t 控制的阶段循环. 最终, 每个节点在多个阶段距离之间选择最小的 $u_t^*(i_t)$ 作为最优决策, 即最短距离. 这样节点 d 计算出了从 s 到它的最短距离.

对图 1 中的有向图, 取定 $s=0, d=5$ 运行上面的算法, 可以得到从 s 到 d 的最短路径, 如图 2 中粗实线所示. 其他节点也计算出了它到 d 的最短路径 (后文中将证明这个结论). 粗虚线则指示了其他节点到 d 的最短路径.

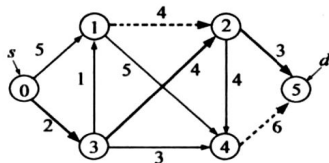


图 2 MFHDR 的计算结果

4.2 路由建立过程

设备通信时的具体路由过程是:

(1) 首先查找本地路由表, 若查找到到目的地的路由表项, 则按照指示的下一跳发送; 否则, 继续 (2).

(2) 开始寻找路径. 首先源节点广播路径选择请求 $RREQ^{[10]}$, 各设备之间采用泛洪的模式传递这个请求, 邻居节点互相传递费用向量 CV .

(3) 决策过程由目的设备节点发起, 计算 $u_0^*(d)$ 并将其值传递给它的邻居设备节点.

(4) 在以后的每一个阶段, 每台设备分别计算 $u_t^*(i_t)$ 和 $A_t^*(i_t)$.

(5) 阶段算法结束时, 各设备节点 (包括源节点) 产生本地决策的下一跳 $f_{opt}(i)$, 在路由表中加入新的路由表项: d 为目的地址, $f_{opt}(i)$ 为下一跳. 初始化此表项路由超时定时器值为 $timemax$, 本次路由选择过程结束.

(6) 各节点定期的广播 HELLO 消息来发现和保持邻居信息, 并记录到它的邻居表中. 一个节点每收到路由表项中 $next$ 节点的 HELLO 消息, 则重新初始化该表项的路由超时定时器为 $timemax$; 各节点随时自动更新 (自减) 其路由表各表项的此定时器的值: 一旦路由表项的定时器为零, 则删除此条路由信息.

5 有效性和复杂度分析

5.1 正确性证明

根据上述算法及过程, 得到两个有关本算法的结论.

结论 1 算法结束后, 每个节点都可以计算出它到目的节点的最短路径.

$$\pi(i) = (i, f_{opt}(i), f_{opt}(f_{opt}(i)), f_{opt}(f_{opt}(f_{opt}(i))), \dots, d).$$

证明: 使用反证法, 对于任一节点 i , 假设算法计算得到 $u_{opt}(i)$ 对应的路径 $\pi(i) = (i, f_{opt}(i), f_{opt}(f_{opt}(i)), f_{opt}(f_{opt}(f_{opt}(i))), \dots, d)$ 不是从 i 到目的节点 d 的最短路径, 也就是说存在一条路径 $\pi_1(i)$ 比路径 $\pi(i)$ 的距离更短, 即 $\pi_1(i)$ 的距离 $u(i)_1$ 小于 $u_{opt}(i)$.

情况 1: 若 $\pi_1(i)$ 与 $\pi(i)$ 阶段数相同: 即两路径中边的条数相同, 那么在算法 1 的步骤 3 求 \min 中选择的应该是 $\pi_1(i)$ 的距离 $u(i)_1$, 这与已知的计算结果: 选择了 $\pi(i)$ 对应的 $u_{opt}(i)$ 矛盾.

情况 2: 若 $\pi_1(i)$ 与 $\pi(i)$ 阶段数不同, 那么在算法 1 的步骤 5: 不同阶段 t 间求 \min 中选择的应该是 $\pi_1(i)$ 的距离 $u(i)_1$, 这与已知的计算结果: 选择了 $\pi(i)$ 对应的 $u_{opt}(i)$ 矛盾.

综上推出假设不成立, 则算法计算得到的各条路径 $\pi(i)$ 是从 i 到目的节点的最短路径.

同时, 可以看到: 算法具有自我学习的能力, 结合路由过程, 各设备能自动生成为本次决定的最短路径服务的路由表项, 并实时维护和更新. 这样, 就保证了算法的可应用性和实时性. 即有如下结论 2.

结论 2 有效性结论. $f_{opt}(i)$ 记录的信息满足需要. 结合路由过程, 即各点使用 $f_{opt}(i)$ 生成供本次路由使用的路由表项是正确的.

证明: 设任意非源非目节点 i , 对应 $f_{opt}(i)$. 此时计算得到的源目节点最短路径为 $\pi(s) = (s, f_{opt}(s), f_{opt}(f_{opt}(s)), f_{opt}(f_{opt}(f_{opt}(s))), \dots, d)$.

情况 1: i 节点不在 $\pi(s)$ 上, 不用讨论此种情况.

情况 2: i 节点在 $\pi(s)$ 上, 那么现在要证明是计算得到的 i 节点到 d 的最短路径

$$\pi(s) = (s, f_{opt}(s), f_{opt}(f_{opt}(s)), f_{opt}(f_{opt}(f_{opt}(s))), \dots, d).$$

..., d) 与 $\pi(s)$ 是重叠的.

已知定理: 最短路径的局部路径也是最短路径.

所以 $\pi(s)$ 的局部路径: 从 i 到 d 的路径也是最短路径. 而 $\pi(i)$ 是从 i 到 d 的最短路径, 所以 $\pi(i)$ 是组成 $\pi(s)$ 的一部分. 所以 $f_{opt}(i)$ 的指示是正确的, 可以作为源目最短路径上的下一跳指示.

5.2 时空复杂度

下面从理论上分析此算法的复杂度. 由上面的路由算法及路由过程, 可以直观的看到, 本算法具有分布式并行计算的特点: 每一个阶段多个设备都参与计算 $u_i^*(i_t)$.

算法的运行阶段: 上限为 $O(|V| - 1)$, 即源目之间可能的最多计算阶段是 $|V| - 1$, 此时得到的路径是跳跃所有节点到达目的节点的路径. 但步骤 4 可以有效的使算法尽早结束循环, 因此迭代次数 $\ll (|V| - 1)$.

单阶段参与计算的设备数目: 如果每个阶段每个设备都参与计算 $u_i^*(i_t)$, 这样会造成所有设备不停的工作, 增加了设备的负担. MFHDR 算法使用了这样的参与条件: $\forall e_{ij} \in E, \exists u_{i-1}^*(j) \neq \infty$, 即只有已计算 $u_{i-1}^*(j)$ 节点的前驱节点计算, 这在算法的代码实现时很容易的提前做出判断. 这样, 每个阶段参与计算的节点数上限是 $O(|V| - 1)$, 而且这些设备是并行工作的.

单阶段单个节点的计算量: 每个节点的入度为 $indegree(i)$, 那么每个阶段单个可工作节点的计算量(扫描后继节点, 求同阶段的 \min) 为 $O(\max indegree(i))$, 对于稀疏拓扑图有 $O(\max indegree(i)) \ll O(|E|)$, 对于非稀疏图有 $O(\max indegree(i)) \leq O(|V| - 1)$, 当全互连时取等号.

空间复杂度: 每个节点要维持一个阶段距离数组和阶段决策数组, 元素数目都是 $|V| - 1$, 记录的是该节点到目的节点经过 t 阶段(跳数)的最短距离以及对应的下一跳. 并维护邻居表和路由表. 数目也分别是 $|V| - 1$. 这就将单点独立计算时源节点要维护的全局拓扑信息分布到各个节点上, 均衡了空间复杂度.

从上述对算法的分析, 可以看到算法的优点: 这种马氏有限多阶段决策解法, 虽然整体上看计算复杂度是 $O(|V| * \max indegree(i))$. 但它的计算工作和拓扑信息分布在了每个设备节点上, 而单阶段中各个节点又是并行工作的, 因此可以减少和均衡各节点的工作负载, 有效利用嵌入式对等互连网络中的各个计算资源. 并且, 对于现实中一般的互连拓扑图, 时间复杂度则远远小于 Dijkstra 等算法上限 $O(|V|^2)$, 甚至远小于 $O(|V||E|)$. 因此, 复杂度的降低、多因素的考虑和分布计算的应用, 使得本算法比传统的 QoS 算法等更适用于嵌入式互连环境.

6 结论

在上面的章节中, 针对嵌入式环境中资源匮乏, 功能专用, 实时性要求高的特点, 利用马尔可夫决策理论, 建立了路由的马氏有限阶段模型, 设计了 MFHDR 路由算法, 并说明了路由的具体执行过程, 证明了算法的正确性和低的时空复杂度. 本算法和路由过程都有效的利用了嵌入式对等网络互连的特点, 具有实时性好, 分布计算, 均衡工作量, 节约单个设备计算和空间资源, 能有效避免环路产生, 对比网络通用的一些路由算法, 它在专用型嵌入式设备互连上是有实际应用价值的.

当然, 研究还有很多有待完善的地方, 比如本文的转移概率采用了 $0-1$ 分布的形式, 节点间信息的更新实现过程还不完善. 在今后的工作中, 我们将进一步完善 MFHDR 算法, 使之更具通用性和实用性, 并将其用于嵌入式通信中间件的实现中. 同时, 我们还将继续沿着序列决策的思路, 研究路由的动态更新, 路由的备份, SPT 树的建立, 多目标决策以及结合传统路由算法的费用预测等问题.

参考文献:

- [1] Elizabeth M Royer, C K Toh. A review of current routing protocols for Ad Hoc mobile wireless networks[J]. IEEE Personal Communications, 1999, 6(2): 46-55.
- [2] Dijkstra E W. A note on two problems in connection with graphs[J]. Numerische Mathematik, 1959, 1: 260-271.
- [3] L Ford. Network Flow Theory[R]. The Rand Corporation, Technical Report P-932, 1956.
- [4] R E Bellman. On a routing problem[J]. Quarterly of Applied Mathematics, 1958, 16(1): 87-90.
- [5] 郑相全, 郭伟, 葛利嘉, 刘仁婷. 一种基于跨层设计和蚁群优化的自组网负载均衡路由协议[J]. 电子学报, 2006, 34(7): 1199-1208.
Zheng Xiangquan, Guo Wei, Ge Lijia, Liu Rentin. A cross layer design and ant colony optimization based load balancing routing protocol for Ad Hoc networks(CALRA)[J]. Acta Electronica Sinica, 2006, 34(7): 1199-1208. (in Chinese)
- [6] Bin Xiao, Qingfeng Zhuge, Edwin H M Sha. Efficient algorithms for dynamic update of shortest path tree in networking[J]. Journal of Computers and Their Applications, 2004, 11(1): 60-75.
- [7] Ljubica Blazevic, Jear Yves Le Boudec, Silvia Giordano. A location based routing method for mobile Ad Hoc networks[J]. IEEE Transactions on Mobile Computing, 2005, 2(4): 97-110.
- [8] Puterman M L. Markov Decision Processes[M]. New York:

John Wiley & Sons, 1994.

[9] 刘克. 实用马尔可夫决策过程[M]. 北京: 清华大学出版社, 2004.

Liu Ke. Applied Markov Decision Processes[M]. Peking: Ts

inghua University Press, 2004. (in Chinese)

[10] C E Perkins, E M Royer, S R Das. Ad hoc On Demand Distance Vector Routing [EB/OL]. IETF RFC 3561. <http://www.ietf.org/rfc/rfc3561.txt?number=3561>, 2003-07.

作者简介:



贾智平 男, 1964 年生于山东即墨, 现为山东大学教授, 在国内外发表学术论文 50 余篇. 主要研究领域为嵌入式系统、分布式计算、实时系统. E-mail: zhipingj@sdu.edu.cn



张承慧 男, 1963 年生于山东德州, 现为山东大学教授, 博士生导师, 主要从事控制理论、工程优化控制研究.



刘甜甜 女, 1983 年生于山东潍坊, 现为山东大学在读硕士生. 主要研究领域为嵌入式系统、分布式计算.



Edwin H.-M. Sha 男, 1964 年生于中国台湾, 现为美国德克萨斯大学达拉斯分校教授, 在国内外发表学术论文 220 余篇, 主要研究领域为并行优化、嵌入式系统.