

多跳无线 Ad hoc 网络的分布式公平调度算法

应玉梅¹, 刘卫², 秦翻¹, 董勤¹

(1. 华中师范大学计算机科学系, 武汉 430079; 2. 华中师范大学计算机网络理论与应用研究中心, 武汉 430079)

摘要: 为兼顾无线 Ad hoc 网络信道资源的公平利用及网络吞吐量, 提出针对多跳无线 Ad hoc 网络的分布式公平调度算法。该算法采用链路竞争图说明网络中各子流的竞争关系, 通过着色算法得到最大无竞争流的集合, 通过邻居节点之间的信息交换得到局部(本地)流的信息, 而无需求全局流的信息交换。实验模拟结果表明, 该算法可以在保证 Ad hoc 网络中各流公平性的同时, 有效提高网络资源的利用率。

关键词: 无线 Ad hoc 网络; 公平调度; 分布式算法; 最大独立子集

Distributed Fair Scheduling Algorithm for Multi-hop Wireless Ad hoc Network

YING Yu-mei¹, LIU Wei², QIN He¹, DONG Meng¹

(1. Department of Computer Science, Huazhong Normal University, Wuhan 430079;

2. Computer Networks Theory and Application Research Center, Huazhong Normal University, Wuhan 430079)

【Abstract】 In order to balance the fairness and throughput of channel resources in wireless Ad hoc network, this paper proposes a distributed fair scheduling algorithm for multi-hop wireless Ad hoc network. The contention relationship of sub-flows are denoted by link contention graph, and the maximal independent sets are formed with coloring algorithm. The algorithm depends on local flow information obtained by information exchange between neighbor nodes. Simulation experimental result proves that the algorithm can efficiently maximize the throughput while maintaining basic fairness among flows.

【Key words】 wireless Ad hoc network; fair scheduling; distributed algorithm; maximal independent subset

1 概述

公平性和资源利用率是无线 Ad hoc 网络设计的 2 个目标, 而这 2 个目标在本质上是冲突的。基于介质共享的多跳无线 Ad hoc 网络公平调度算法的设计存在着巨大挑战, 因为: (1)数据流既存在时间领域的竞争也有空间领域的竞争。(2)由于 Ad hoc 网络本身的分布式特性, 公平调度只能依靠局部信息来完成, 但局部公平并不意味着全局公平。(3)无线 Ad hoc 网络的资源数目有限, 如信道容量和节点能量。

无线 Ad hoc 网络已有的研究成果主要采用 2 种方法实现公平调度: 基于效用函数的方法和保证最小公平性的方法。对于第 1 种方法, 用户的需求往往很难用精确的函数表示^[1], 本文沿用第 2 种方法。之前许多研究^[1-5]都集中在保证基本公平性或者提高网络的吞吐量方面。文献[2]提出了一个理想的集中式公平调度模型, 该模型假定调度器可以获得整个网络中各流的精确信息, 并以集中式公平调度模型为依据, 设计了一个分布式的算法以近似该集中式模型。但是该算法具有很高的复杂性且只针对单跳网络。文献[3]提出了一个低复杂度分布式公平调度算法, 但当网络中的节点移动速率比较高时, 该算法很不稳定。文献[4]提出了一种新的预测方法, 它基于上下限的方法分析并保证了公平性, 但该算法是集中式而非分布式的。基于以上分析, 本文提出了针对多跳无线 Ad hoc 网络的分布式公平调度算法, 目标是在保证各个流公平性的同时有效地提高网络吞吐量。

2 系统模型

2.1 网络模型

无线 Ad hoc 网络用有向图 $G(V,E)$ 表示, 其中, V 表示网

络中的节点; E 表示边的集合。如果网络中的节点 u 和节点 v 在各自的最大覆盖范围之内, 则边 $(u,v) \in E$ 。假定网络中所有节点的最大覆盖范围相同并且所有流共享一个容量为 C 的单独物理信道。本文仅考虑各个流之间的空间竞争, 假定在时间上是没有冲突的。对于多跳的 Ad hoc 网络, 2 跳内的流均存在竞争。

2.2 带权链路竞争图

本文采用带权链路竞争图^[5]表示相互竞争流的空间竞争关系。一个单独物理链路上传输的数据流称为子流^[5], 并认为如果 2 个子流中有一个子流的源端或目的端在另一个子流的源端或目的端, 则这 2 个子流存在竞争。图 1 是一个随机的无线 Ad hoc 网络拓扑图实例, 其中有 7 个节点 {1, 2, 3, 4, 5, 6, 7} 以及 3 个流 $\{f_1, f_2, f_3\}$, 与该网络相对应的链路竞争图见图 2, 其中, 顶点表示连接节点的网络链路(子流)。

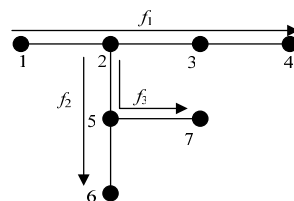


图 1 无线 Ad hoc 网络

基金项目: 国家自然科学基金资助项目(60473085)

作者简介: 应玉梅(1984-), 女, 硕士研究生, 主研方向: 无线 Ad hoc 网络, 智能控制; 刘卫, 硕士研究生; 秦翻、董勤, 本科生
收稿日期: 2008-07-10 **E-mail:** candy_yumei@yahoo.com.cn

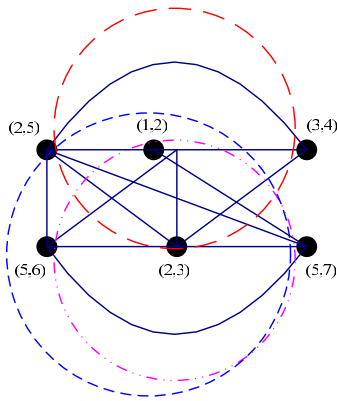


图 2 基于图 1 的 3 个最大竞争链路图

图 2 中的各个节点只要有边相连, 就表示节点所对应的子流存在着竞争。所有相互竞争的流的带宽总和上限是信道带宽 C , 满足此条件是可行带宽分配的前提条件。可得到图 1 的 3 个最大竞争链路图 $\{l_{12}, l_{23}, l_{25}, l_{34}\}$, $\{l_{12}, l_{23}, l_{56}, l_{57}\}$, $\{l_{12}, l_{23}, l_{25}, l_{56}, l_{57}\}$ 。

每一个最大竞争链路图中所有的子流都是相互竞争的。用 b_{ij} 表示链路 l_{ij} 分配到的带宽, 则有如下的资源限制:

$$b_{12} + b_{23} + b_{25} + b_{34} \leq C$$

$$b_{12} + b_{23} + b_{56} + b_{57} \leq C$$

$$b_{12} + b_{23} + b_{25} + b_{56} + b_{57} \leq C$$

为网络中的各流事先分配一个权值 $w(i)$, 且规定属于同一个流的各个子流具有相同的权值。

2.3 公平性定义

无线多跳 Ad hoc 网络的公平性是一个与拓扑相关的属性^[5]。如果满足资源上限 $b_1 + b_2 + \dots + b_n \leq C$, 则对最大竞争链路图中相互竞争的流的带宽分配策略 (b_1, b_2, \dots, b_n) 是可行的。这是保证公平性最基本的前提, 另外对网络的公平性加强了限制条件, 在任意时间段 $[t_1, t_2]$ 内, 对 2 个相互竞争的流 i 与 j , 满足: $|b_i/w(i) - b_j/w(j)| < \varepsilon$, 其中, $b_i/w(i)$ 为流 i 在单位权值上获得的带宽, 保证各个流获得长期的公平性。

3 分布式的公平调度算法

3.1 最大独立子集的确定

链路竞争图中每个最大独立子集表示一个最大的、可区分的无竞争链路集合。本文通过改进距离-2(D-2)着色算法^[5]求链路竞争图的最大独立子集。该算法是周期轮转同步执行的。每轮分为 2 个阶段:

(1) 每个被激活但未着色的顶点(以 u 为例)为自己选择一个尝试颜色 $color(u) = k$ 。然后选择颜色成功的顶点向其所有邻居节点广播其当前尝试信息 $\{ID, color(u)\}$ 。

(2) 每个顶点首先判断在第(1)阶段所选择的尝试颜色是否可行。顶点 u 在 2 跳之内的邻居节点所选择的颜色未与其颜色相同, 则颜色 $color(u) = k$ 可行^[5]。此时 $color(u) = k$ 成为顶点 u 的永久颜色, 顶点 u 向邻居节点广播成功信息 $\{ID, color(u)\}$ 。至此该轮选色结束。

接着分布式地执行相同的步骤, 在下一轮中已经成功选色的节点不再参与。最后, 所选颜色相同的节点属于同一个集合, 即最大独立子集。

3.2 公平调度算法

本文采用基于排队时间的公平排队算法(STFQ)^[4]对流进行调度。为刚到达的数据包设置 2 个标志: 开始标志和结束

标志。流 i 的序号为 n 的数据包的到达时间表示成 $A(t_i, n)$, 开始时间标志 $s_{i,n}$ 和结束标志 $f_{i,n}$ 为

$$s_{i,n} = \max\{V(A(t_i, n), f_{i,n-1})\}$$

$$f_{i,n} = s_{i,n} + L/w(i)$$

其中, L 为数据包的大小。调度器在时刻 t 处的虚拟时间 $V(t)$ 设置为: 如果此时信道繁忙, 则 $V(t)$ 为当前时刻接收服务的数据包的开始时间; 否则, $V(t)$ 为已经服务的包的最大结束时间。当流 i 的一个包 p 在 t 时刻到达, 它被赋予标志 $s_{i,n}$ 。队列中的数据包包按开始标记递增的顺序接收服务。

3.3 算法描述

整个算法的过程描述如下, 网络中的每个节点分布式地执行该算法:

Input graph $G = (V, E)$, weights $w(i)$ for each flow

1: Form the link contention graph based on the given network model $G = (V, E)$.

2: For $i=1$ to N do

for $j=1$ to M do

Compute $s_{i,j}$, and $f_{i,j}$.

end for

End For

3: Pre-fill the scheduling window with packets from the scheduling queue in increasing order of start tag.

/*iteration loop*/

4: While scheduling queue is not empty

Re-fill the scheduling window with packets from the scheduling queue in increasing order of start tag.

SUM(b_k) $\leq C$ /* apply the fairness constraints */

$f_p = f$ with the least start tag /*grants transmission priority */

5: D2-coloring algorithm /*apply D2-coloring algorithm */

while(node u is active and has no color)

color(u) = k /* Choose a color for node u randomly.*/

if (($\{ID, color(u)\}$) does not conflict within the 2-hops)

color(u) = k

node u broadcast $\{ID, color(u)\}$ to its neighbor nodes and make k be the permanent color for node u .

End while.

6: Matching the appropriate set which contains the flow f_p .

7: Transmit the related packets simultaneously.

End While

3.4 算法的复杂性分析

分布式 D-2 着色算法的运行复杂度为 $O(\Delta \log^2 n)$, 着色问题需要用到的颜色数目为 $O(\Delta)$ 。本算法的运行时间主要取决于 D-2 着色算法, 因此, 整个算法的时间复杂度为 $O(\Delta \log^2 n)$ 。

4 仿真与结果分析

本文通过仿真实验验证所提出的公平调度算法的性能。信道总容量 C 设为 2 Mb/s。仿真场景中节点的数目为 M , 网络中流的数目为 n 。对每个流的开始节点和结束节点的选取是任意的。

图 4 给出了本文算法和 STFQ 算法的公平性指标随网络中流数目的增加而变化的对比, 这里引入了一个公平性指标 (fairness index):

$$F_{\text{fairness}} = \frac{(\sum_f T_f / w(f))^2}{N \times \sum_f (T_f / w(f))^2}$$

其中, T_f 为流 f 的吞吐量; $w(f)$ 为对应流的权值。网络中流的

数目 m 设置为(5,10,15,20,25,30),观察该公平性指标随着流数目增加的变化情况。由图 3 可以得出,本文的算法能够很好地满足公平性,并随网络中流数量的增加,其公平性与 STFQ 算法越来越接近。

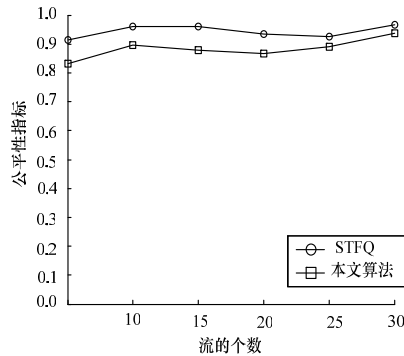


图 3 公平性指标比较

图 4 验证了网络中各子流间的公平性,其中设置了 12 个流,子流数目为 30,采用单位权值获得的带宽 $b_i/w(i)$ 验证。从图 4 中可知,各个子流具有相近的 $b_i/w(i)$,满足 2.3 节中提出的公平标准,具有长期的公平性。图 5 给出了网络总吞吐量随流数目的增加而变化的情况,其参数设置同图 3。与 STFQ 相比,本文算法在吞吐量上有显著的提高。综上所述,本文的分布式公平调度算法具有与 STFQ 算法相近的公平性,且在保证公平性的同时,较 STFQ 算法在网络吞吐量方面有显著的提高。

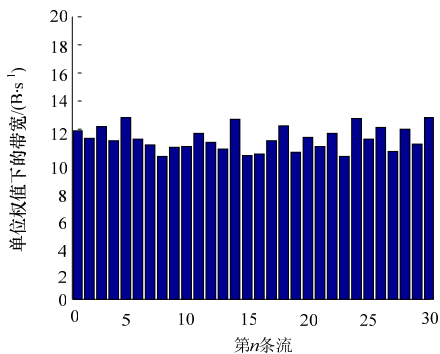


图 4 各子流之间的公平性

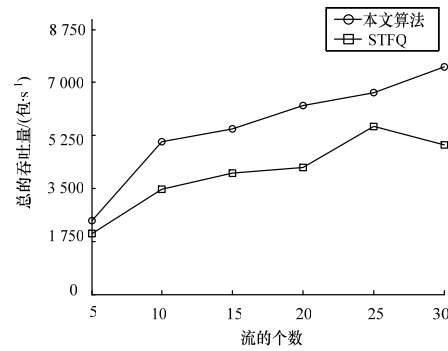


图 5 网络总的吞吐量比较

5 结束语

本文从一个新的角度定义网络的公平性,在保证公平性的限制条件下,采用链路竞争图表示网络中子流的竞争关系,并通过着色算法将网络中的子流分配到各个相互独立的最大无竞争的集合中,使得属于同一最大独立子集的流尽可能地同时调度,从而提高网络的吞吐量。

参考文献

- [1] Li Baochun. End-to-end Fair Bandwidth Allocation in Multi-hop Wireless Ad hoc Networks[C]//Proc. of IEEE ICDCS'05. [S. l.]: IEEE Press, 2005.
- [2] Xue Yuan, Li Baochun, Nahrstedt K. Price-based Resource Allocation in Wireless Ad hoc Networks[C]//Proceedings of International Workshop on Quality of Service. [S. l.]: ACM Press, 2003.
- [3] Luo Haiyun, Lu Songwu. A Self-coordinating Approach to Distributed Fair Queueing in Ad hoc Wireless Networks[C]//Proceedings of IEEE INFOCOM'01. [S. l.]: IEEE Press, 2001.
- [4] Goyal P, Vin H, Chen Haicheng. Start-time Fair Queueing: A Scheduling Algorithm for Integrated Service Access[C]//Proceedings of ACM SIGCOMM'96. [S. l.]: ACM Press, 1996.
- [5] Srinivasan P, Rajiv G. Distributed Algorithms for Coloring and Domination in Wireless Ad hoc Networks[C]//Proceedings of the 24th Conference on Found of Software Technology and Theoretical Computer Science. [S. l.]: Springer Press, 2004.

编辑 张帆

(上接第 97 页)

参考文献

- [1] Floyd S, Kevin F. Promoting the Use of End-to-end Congestion Control in the Internet[J]. IEEE/ACM Transactions on Networking, 1999, 7(4): 458-472.
- [2] Misra V, Gong Weibo, Towsley D. Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED[C]//Proceedings of IEEE ACM SIGCOMM. Stockholm, Sweden: IEEE Press, 2000.

- [3] Gao Weibing, Wang Yufu, Homaifa A. Discrete-time Variable Structure Control System[J]. IEEE Transactions on Industrial Electronics, 1995, 42(2): 117-122.
- [4] Quet P F, Özbay H. On the Design of AQM Supporting TCP Flows Using Robust Control Theory[J]. IEEE Transactions on Automatic Control, 2004, 49(6): 1031-1036.

编辑 顾姣健