DOI:10.19789/j.1004-9398.2018.04.007

无线自组织网络的改进MGD分组调度算法*

涂 朴[1,2] 赵全军[2] 李 斌[1,2]

(1. 四川文理学院达州智能制造产业技术研究院,四川 达州 635000; 2. 四川文理学院智能制造学院,四川 达州 635000)

摘 要

随着无线自组织网络研究和应用的发展,人们对它提出了更高的要求. 该文在原有MGD(modified Galois field design)算法的基础上提出了一种改进算法,并对改进算法的最小吞吐量、最大时延进行了理论推导,对改进算法的性能参数进行了数值计算分析. 分析结果显示:改进算法在吞吐量、时延等方面都具有较好的优越性,为实际系统的设计和未来网络研究提供参考.

关键词: 无线自组织网络,分组调度,MGD算法,吞吐量,延迟.

中图分类号: TN929.5

0 引 言

随着计算机网络技术与无线通信技术的快速发展,无线自组织网络也越来越受到人们的重视,其网络性能的好坏一直是人们关注的对象和学术界研究的热点. 其中,分组调度机制是无线自组织网络构造所依赖的最主要的性能保证机制之一,而其核心问题是分组调度算法问题. 设计出有效的分组调度算法,研究人员进行了大量的研究. 过去的研究工作中,公平分配带宽和最大资源利用率被认为是两个最重要的设计目标. 在文献[1-3]中仅考虑资源利用率的分组调度,但是缺少考虑在传输过程中最大吞吐率的保证. 在文献[4-6]中,研究者们主要从效益函数(utility function)的角度,以及设计公平队列等方法来保证分组调度过程中的公平性问题. 以上研究调度算法在设计时要么仅考虑资源利用率,要么仅考虑公平需求,这在某些应用背景下的确是合理的. 但是存在较大的应用局限性. 而基于二者折中的一些研究算法[7-9],其思想主要是首先保证每个流都有一个最小的公平份额,在此基础上实现资源的最大利用. 作者在MGD算法[10]的基础上,提出了改进算法,其思想是一种拓扑透明的链路激活性算法,它的基本思想是通过给每个节点分配一个唯一的多项式,保证节点在一帧中至少有一个时隙与某一邻居节点进行数据通信,从而提高了网络的吞吐量,降低了网络时延,提升了网络整体性能.

1 MGD算法改进模型

在改进的MGD算法中,对于一个确定的节点 $u \in V$,其邻居节点的集合用 N_u 表示,$|N_u|$ 表示节点 u 的邻居数. 当节点 u 向其任意一邻居 v 发送分组时,如果节点 v 或者 v 的除 u 以外的任意一邻居也在同一时隙发送分组数据,就会导致节点 u 传输失败. 因此我们将 $N_v \cup \{v\} - \{u\}$ 定义为节点 u 的干扰节点集,且存在 $|N_v \cup \{v\} - \{u\}| = |N_v| \leqslant D_{\max}$.

我们定义一帧的时隙集为 Ω,且 $|\Omega| = pq$. 一帧中系统分配给节点 u 的时隙集我们用 A_u 表示,显然 $|A_u| = q$. A_u 可分为两个部分:系统分配给节点 u 并且向其邻居成功传输分组的时隙,用 S_u 表示;系统分配给节点 u 但是产生冲突的时隙,用 C_u 表示. 显然,$|S_u| + |C_u| = |A_u| = q$. 在一帧中分配给 u 的干扰节点集 $N_v \cup \{v\} - \{u\}$ 的时隙用 I_u 表示. 没有分配给节点 u 及其干扰节点 $N_v \cup \{v\} - \{u\}$ 的时隙(即能够被 u 复用的时隙)我们用 R_{uv} 表示,且 $R_{uv} = \Omega - A_u \cup I_u$.

为了更大限度地提高网络吞吐量,在改进算法

收稿日期:2017-05-22

* 达州市科技局惠民计划项目(KJJ2015003);四川文理学院校级科研项目(2016KZ009Y);四川省教育厅项目(18ZB0508).

(C)1994-2020 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

中引入了 IEEE 802.11 四步握手 RTS/CTS 机制[11].

根据 RTS/CTS 机制的握手过程,我们将 MGD 算法的每个时隙再细分成四个阶段: RTS(请求传输阶段)、CTS(应答阶段)、DATA(数据传输阶段)以及 ACK(确认阶段). 改进算法的 TSMA 帧结构如图 1 所示.

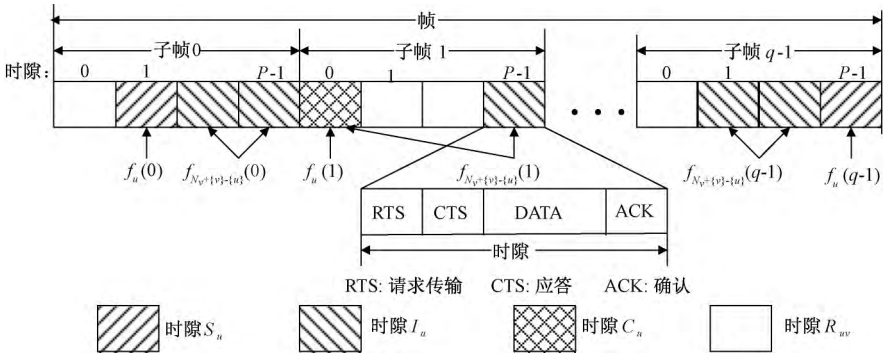


图 1　改进算法的 TSMA 帧结构

改进算法的握手机制和 IEEE 802.11 的四步握手过程相似. 下面将结合握手机制,针对 MGD 算法的两个时隙浪费情况,对其进行相应的改进.

2　对时隙 C_u 部分的改进

在这种情况下,时隙的浪费是由于干扰节点集 $N_v \cup \{v\} - \{u\}$ 产生的. 冲突的产生使得同时发送分组的两个节点都不能成功传输. 系统的吞吐量受到了制约. 为了解决干扰节点集 $N_v \cup \{v\} - \{u\}$ 产生的冲突,使时隙 C_u 得到利用,我们引入了概率 $p_n(0 \leqslant p_n \leqslant 1)$.

对于任意的节点 u,如果它要在时隙 $i \in A_u$ 向节点 v 发送分组数据,首先节点 u 在时隙 i 的 RTS 阶段向接收节点 v 发送 RTS 分组,表明 u 要向 v 发送数据,并预约时隙 i. 如果节点 v 成功接收 RTS 分组,它将回复 CTS 分组给 u. 通过是否成功接受 CTS 分组,节点 u 将知道当前时隙是否产生了冲突,也就是知道了当前时隙是属于 S_u 还是 C_u. 如果当前时隙属于 S_u,那么说明节点 u 在传输 RTS 分组时没有受到节点集 $N_v \cup \{v\} - \{u\}$ 的干扰. 此时时隙 i 预约成功,并且节点 u 能够向节点 v 成功传输分组数据. 在分组数据传输完之后,节点 v 回复 ACK 给节点 u,RTS/CTS 握手过程完毕; 如果当前时隙属于 C_u,那么说明 u 向节点 v 发送 RTS 分组时,受到了节点集 $N_v \cup \{v\} - \{u\}$ 的干扰. 在这种情况下,改进算法将以某一概率来利用当前时隙 C_u. 具体来说就是,如果节点 u 没有接收到 v 发来的 CTS 分组,它将认为时隙预留失败,并且以概率 p_n 发送分组数据. 同样与 u 发生碰撞的节点也意识到时隙预约失败,它也将以概率 p_n 进行分组传输.

通过以上改进,碰撞时隙 C_u 得到了利用,算法的吞吐量得到了较大的提高.

3　对时隙 R_{uv} 部分的改进

前面定义了时隙 R_{uv} 为一帧中没有分配给节点 u 及其干扰节点集 $N_v \cup \{v\} - \{u\}$ 的时隙. 如果 u 在时隙 R_{uv} 发送分组,对其邻居节点将不会产生影响. 但对于网络中任意的一个节点,它只知道系统分配给它的时隙 A_u,而不会知道时隙 R_{uv} 的存在. 因此,为了在 $\Omega - A_u$ 的时隙集中成功地发现时隙 R_{uv},并对其中的空闲时隙加以利用,我们对算法也进行了改进.

当节点 u 处于未分配给它的时隙 $\Omega - A_u$ 时,它将侦听信道的空闲情况. 如果侦听到信道空闲,节点 u 认为在当前时隙它的邻居没有接收分组数据,可以进行数据的传输. 当然,如果其他节点也侦听到这个时隙为空闲时隙,那么它们也可以在这个时隙传输分组,这样就会和节点 u 产生冲突. 因此,为了避免冲突的发生,只要 u 在时隙 $\Omega - A_u$ 中侦听到信道空闲,它将以概率 p_n 传输分组数据. 显然通过以上改进,我们能发现当前时隙是否属于 R_{uv},并且对其以概率 p_n 加以利用.

应当指出的是,使用 RTS/CTS 握手机制会使整个网络的效率有所下降. 但 RTS 分组与 CTS 分组的长度很短,与数据帧相比开销不算大,对网络开销的增加可以忽略. 相反,若不使用 RTS/CTS 握手机制,则一旦发生碰撞而导致数据帧重发,则浪费的时间就更多.

(C)1994-2020 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

以上对 MGD 算法的改进是拓扑透明的,不需要特殊的拓扑信息,不会增加额外的控制负载给网络带来更大的开销. 概率 p_n 是一个简单的公共参数,它对于每个节点都是相同的. 算法的改进解决了干扰节点产生的碰撞问题以及空闲时隙的利用,吞吐量有了明显的提高,从而使得系统性能有了明显的改善.

4 改进算法的性能分析

下面同样根据文献[1]中对 Grand 算法最小吞吐量、最大传输时延和最小传输时延的推导方法,对改进算法的性能进行推导. 在推导过程中,由 RTS/CTS 握手机制产生的负载忽略不计.

在一帧中节点 u 与其干扰节点 $N_v \cup \{v\} - \{u\}$ 的最大碰撞次数为 $kD_{\max}$,那么

$$|C_u| = |A_u \cap I_u| \leqslant kD_{\max}. \tag{1}$$

我们知道:

$$|R_{uv}| = pq - |A_u \cup I_u|, \tag{2}$$

其中,$|A_u \cup I_u|$为系统分配给节点 u 及其干扰节点 $N_v \cup \{v\} - \{u\}$ 的时隙. 由于:

$$|A_u \cup I_u| \leqslant (|N_v| + 1)q, \tag{3}$$

所以:

$$\begin{aligned} |R_{uv}| &\geqslant pq - (|N_v| + 1)q \\ &\geqslant pq - (D_{\max} + 1)q \\ &= pq - D_{\max}q - q. \end{aligned} \tag{4}$$

又因为网络中的每个节点必须至少有一个邻居节点和一个干扰节点,并且在一帧中每个节点都必须至少有一个时隙成功向其邻居发送分组数据,因此:

$$|R_{uv}| \leqslant pq - q - 2. \tag{5}$$

在时隙 S_u 中,节点 u 能够成功地向其邻居节点 v 传输分组数据,其中 $|S_u| = q - |C_u|$. 而在时隙 C_u 中,u 将以概率 p_n 进行传输. 如果用 p_g 表示在 C_u 中成功传输分组的概率,那么:

$$p_g = p_n(1 - p_n)^{|N_v \cup \{v\} - \{u\}|}, \tag{6}$$

其中,p_n 是节点 u 在 C_u 中传输的概率,$(1 - p_n)^{|N_v \cup \{v\} - \{u\}|}$是节点集 $N_v \cup \{v\} - \{u\}$ 在 C_u 中不传输的概率. 在 R_{uv} 中成功传输分组的概率 p_g 也等于(6)式.

改进算法的 T(即每个节点在一帧中能够成功传输分组的数目)为:

$$\begin{aligned} T &= |S_u| + p_g|C_u| + p_g|R_{uv}| \\ &= q - |C_u| + p_g|C_u| + p_g|R_{uv}| \\ &= q - (1 - p_g)|C_u| + p_g|R_{uv}|, \end{aligned} \tag{7}$$

其中,$|S_u| = q - |C_u|$是分配给节点 u 没有发生碰撞的时隙数,$p_g|C_u|$和 $p_g|R_{uv}|$分别是 u 在 C_u 和 R_{uv} 中成功传输的时隙数. 将(1)和(2)式带入得到:

$$\begin{aligned} &T = q - (1 - p_g)|C_u| + p_g|R_{uv}|, \\ &q + p_g(pq - q - 2) \geqslant T \geqslant q - (1 - p_g)kD_{\max} + \\ &\qquad p_g(pq - qD_{\max} - q), \end{aligned} \tag{8}$$

即 $T_{\min} = q - (1 - p_g)kD_{\max} + p_g(pq - qD_{\max} - q)$,$T_{\max} = q + p_g(pq - q - 2)$. 由定义 6 得到最小吞吐量 $G_{\min}$ 为:

$$\begin{aligned} G_{\min} &= T_{\min}/L \\ &= [q - (1 - p_g)kD_{\max} + p_g(pq - qD_{\max} - q)]/pq, \end{aligned} \tag{9}$$

将式(2) $q = kD_{\max} + 1$ 带入上式:

$$\begin{aligned} G_{\min} &= \frac{kD_{\max} + 1 - (1 - p_g)kD_{\max} + p_g[p(kD_{\max} + 1) - D_{\max}(kD_{\max} + 1) - (kD_{\max} + 1)]}{p(kD_{\max} + 1)} \\ &= \frac{(kD_{\max}p + p - kD_{\max}^2 - D_{\max} - 1)p_g + 1}{p(kD_{\max} + 1)}. \end{aligned} \tag{10}$$

为了求最大最小吞吐量,我们通过解方程 $\partial G_{\min}/\partial p_g = 0$ 得:

$$p_n = 1/(1 + D_{\max}). \tag{11}$$

因此,当 $p_n = 1/(1 + D_{\max})$ 时,$G_{\min}$ 取得最大值. 一旦我们知道了网络的最大度 $D_{\max}$,就能很容易地得到最优的传输概率 p_n 为多少.

由定义 7 得到改进算法的最大传输时延为:

$$\begin{aligned} DT_{\max} &= L/T_{\min} \\ &= pq/[q - (1 - p_g)kD_{\max} + p_g(pq - q - qD_{\max})] \\ &= p(kD_{\max} + 1)/[kD_{\max}p + p - kD_{\max}^2 - D_{\max} - 1)p_g + 1]. \end{aligned} \tag{12}$$

5 仿真结果及分析

为了更好地说明改进算法的优良性能,我们通过数值计算对 MGD 算法和改进算法进行分析比较.

同样通过 MATLAB 来进行计算验证. 其他参数设定如下: 当 $p_n = 1/(1 + D_{\max})$ 时,改进算法的最小

(C)1994-2020 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

吞吐量达到最大值. 为了计算改进算法的最小吞吐量，我们取 $p_n = 1/(1 + D_{\max})$，最大度 $D_{\max}$ 从 4 增加到 42.

算法的最小吞吐量、最大传输时延和最小传输时延是考察的目标. 为了检验吞吐量和时延相当于节点数目的变化，通过在节点数目分别等于 100、500 以及 1 000 的情况下对 MGD 算法和改进算法的性能进行数值计算比较.

1) 最小吞吐量的比较

由图 2、图 3 和图 4 可以看出，随着节点的度 $D_{\max}$ 的增加，最小吞吐量曲线急剧下降，这说明节点度数增加对网络吞吐量的影响很大. 由图 2 可以看出，在节点最大度 $D_{\max}$ 开始增加的一段范围内，改进算法的最小吞吐量明显高于 MGD 算法. 随着 $D_{\max}$ 的增加 MGD 算法和改进算法的曲线逐渐接近. 当 $D_{\max}$ 达到一定值后，两者的最小吞吐量都下降到很小值. 由图 3 和图 4 看出，当节点数目 N 为 500 和 1 000 时，当 $D_{\max}$ 的值相对较小时，改进算法的最小吞吐量明显大于 MGD 算法. 这说明当 N 达到一定值后，改进算法的性能明显优于 MGD 算法. 这是因为: MGD 算法存在两方面的时隙浪费(包括分配给节点并与其它节点发生碰撞的时隙 C_u 以及没有分配给任何节点的时隙 R_{uv})，改进算法中对这两种情况都以概率 p_n 进行利用，并且帧长度较小，因此改进算法的整体吞吐量高于 MGD 算法; 当 $D_{\max}$ 不变化时，随着网络中节点数目 N 的增加，MGD 算法的最小吞吐量有所下降，如图 2、图 3 和图 4 中当 $D_{\max} = 10$时，MGD 算法的最小吞吐量由大到小依次是: $N = 100$、$N = 500$ 和 $N = 1\ 000$. 而改进算法的最小吞吐量随着节点数目 N 的增加，性能越来越优.

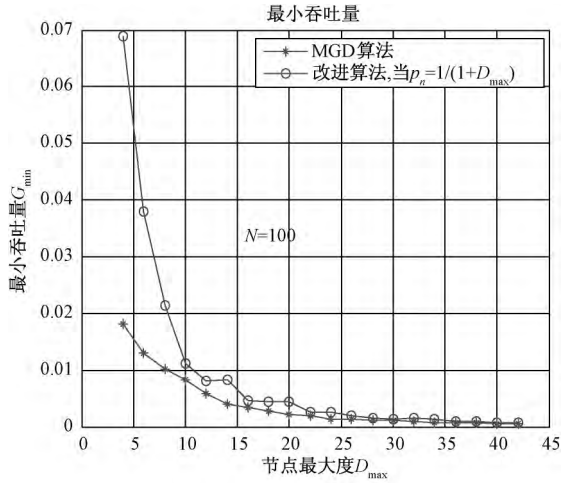


图 2　$N = 100$ 时最小吞吐量的比较

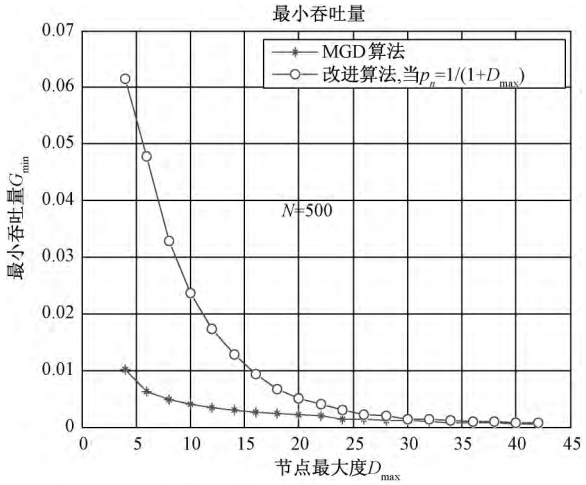


图 3　$N = 500$ 时最小吞吐量的比较

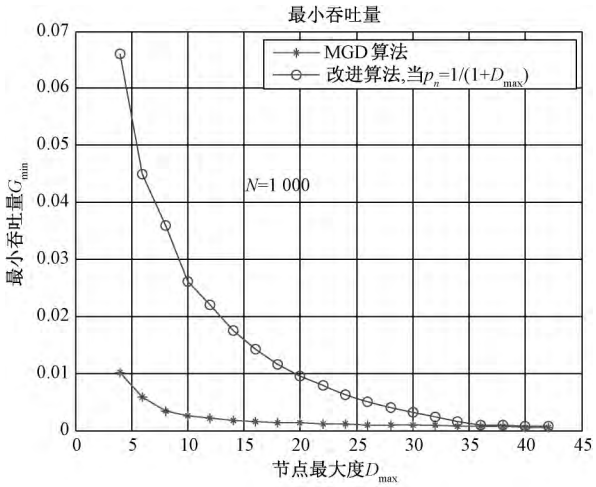


图 4　$N = 1\ 000$ 时最小吞吐量的比较

如图 5 中，$D_{\max}$ 在 5 到 30 之间时，三条曲线对应的值从大到小依次是: $N = 1\ 000$、$N = 500$ 和 $N = 100$. 这说明网络节点数目 N 对网络的吞吐量也有较显著的影响.

2) 最大时延的比较

由图 6 和图 9 可以看出，随着节点最大度 $D_{\max}$ 的增加，最大时延曲线不断上升，这说明节点邻居数目增加对网络时延的影响很大; 由图 6 可以看出，当 $N = 100$ 时，MGD 算法和改进算法的最大时延曲线比较接近，但改进算法的最大时延始终小于 MGD 算法，且随着 $D_{\max}$ 的增加，两者差距越来越大. 由图 7 和图 8 可以看出，随着节点数 N 的不断增加，改进算法的最大时延远远小于 MGD 算法. 这说明当 N 值一定后，改进算法的性能明显优于 MGD 算法. 这也是由 MGD 算法存在两方面的时隙浪费所引起的. 图 9 主要考察的是不同 $D_{\max}$ 值的情况下三种不

(C)1994-2020 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

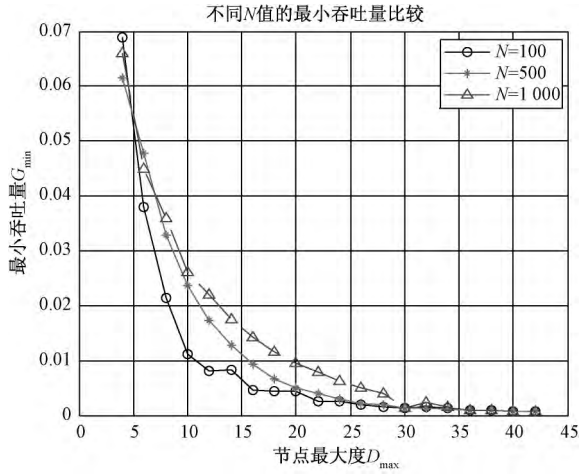


图5 不同 N 值的改进算法最小吞吐量比较

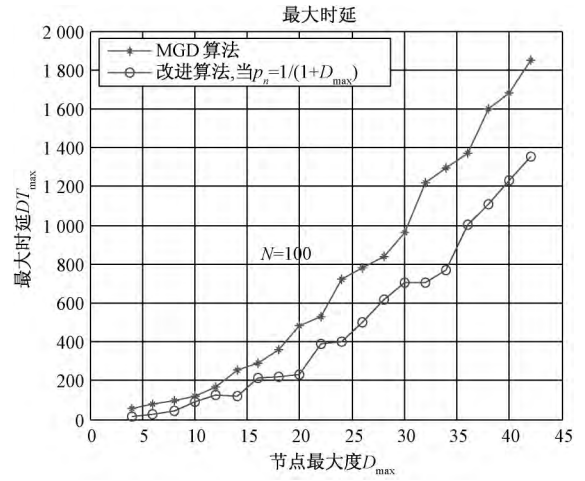


图6 $N=100$ 时的最大时延比较

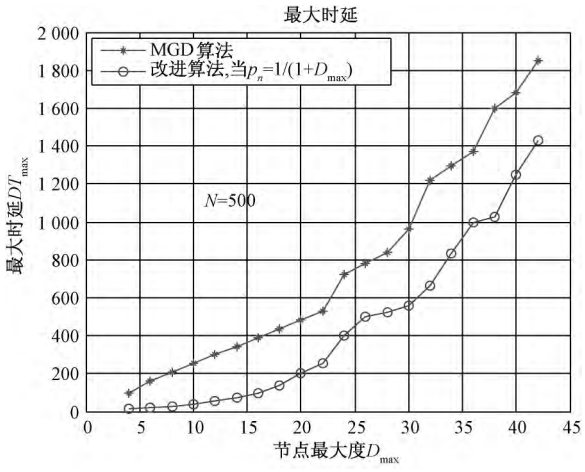


图7 $N=500$ 时的最大时延比较

同网络规模下的延迟问题,当 D_{max} 固定时,随着节点

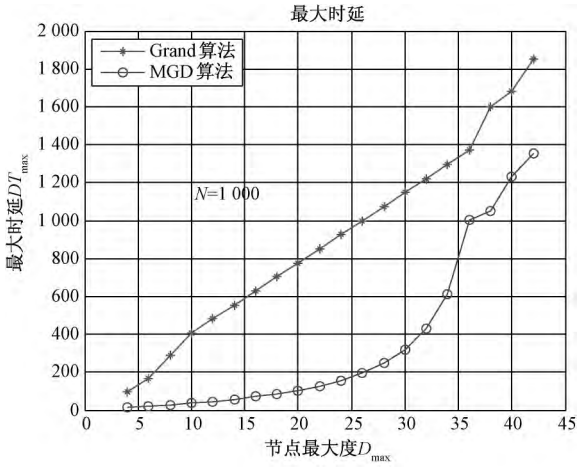


图8 $N=1\ 000$ 时最大时延比较

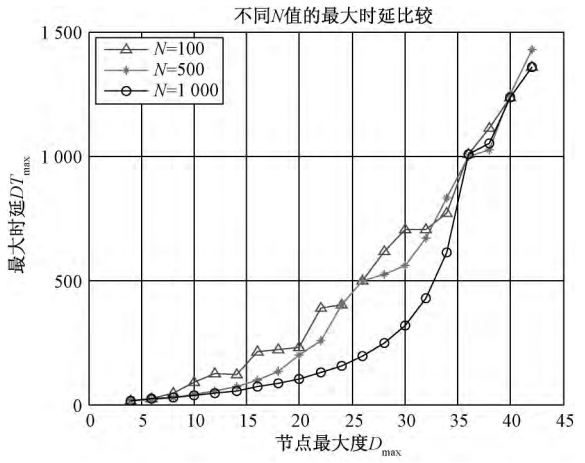


图9 不同 N 值的最大时延比较

数目 N 的增加,改进算法的最大时延值有所下降,如图9中所示,当 $N=1\ 000$ 时的最大时延值小于 $N=100$时的值. 改进的算法对规模较大的网络的延迟性能有一定的改善. ”

6 总 结

本文对 MGD 算法进行了改进,并通过模拟计算对 MGD 算法以及改进算法进行比较分析. 为了考察改进算法的性能,分别在 $N=100$ 和 $N=1\ 000$ 两种情况下进行数值计算. 并得出本文的改进算法最大吞吐量、最大时延方面与 MGD 算法进行比较,有较大的改善. 改进算法引入了 RTS/CTS 机制的握手过程,针对 MGD 算法的两种时隙浪费情况进行相应的改进,从整体性能来看,这种改进是成功的.

(C)1994-2020 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

参 考 文 献

[1] Ju J H, Li V O K. An optimal topology-transparent scheduling method in multihop packet radio networks [J]. IEEE/ACM Transactions on Networking, 1998, 6(3): 298 – 306.

[2] Ephremides A, Truong T V. Scheduling broadcasts in multihop radio networks [J]. IEEE Transactions on Communications, 1990, 38(4): 456 – 460.

[3] Kanodia V, Li C, Sabharwal A, et al. Distributed priority scheduling and medium access in ad hoc networks [J]. Wireless Networks, 2002, 8(5): 455 – 466.

[4] Nandagopal T, Kim T E, Gao X, et al. Achieving MAC layer fairness in wireless packet networks [C]. Boston MA USA: International Conference on Mobile Computing & Networking, 2000: 87 – 98.

[5] Huang X L, Bensaou B. On max-min fairness and scheduling in wireless ad-hoc networks: analytical framework and implementation [C]. Long Beach, CA USA: ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2001: 221 – 231.

[6] Salarian H, Khadivi P. Fair data flows scheduling schema for multihop wireless ad hoc networks [C]. Sails Auckland, New Zealand: IEEE Wireless Communications, 2010: 503 – 508.

[7] Luo H Y, Lu S W, Bharghavan V. A new model for packet scheduling in multihop wireless networks [C]. Boston MA USA: International Conference on Mobile Computing and Networking, 2000: 76 – 86.

[8] Luo H Y, Lu S W. A topology-independent fair queueing model in ad hoc wireless networks [C]. Osaka Japan: 2012 20th IEEE International Conference on Network Protocols (ICNP), 2000: 325.

[9] Kanodia V, Li C Z, Sabharwal A, et al. Ordered packet scheduling in wireless ad hoc networks: mechanisms and performance analysis [C]. EPFL Lausanne Switzerland: MobiHoc´02 Proceedings of the 3rd ACM International Symposium on Mobile ad Hoc Networking & Computing, 2002: 58 – 70.

[10] Cai Z J, Lu M, Georghiades C. Topology-transparent time division multiple access broadcast scheduling in multihop packet radio networks [J]. IEEE Transactions on Vehicular Technology, 2003, 52(4): 970 – 984.

[11] 及晓梅. 关于 IEEE 802.11 协议 MAC 层 DCF 机制的研究与改进 [D]. 天津: 天津大学, 2004.

Improved MGD Packet Scheduling Algorithm Based on Wireless Ad Hoc Networks

Tu Pu[1,2]　Zhao Quanjun[2]　Li Bin[1,2]

(1. Dazhou Industrial Technology Institute of Intelligent Manufacturing, Sichuan University of Arts and Science, Dazhou Sichuan 635000;
2. School of Intelligent Manufacturing, Sichuan University of Arts and Science, Dazhou Sichuan 635000)

Abstract

With the development of the research and application in Ad hoc, people wanted to introduce multi-media traffic into it. An improved scheduling algorithm based on the original modified Galois field design algorithm was presented. The minimum throughput and maximum delay of the improved scheduling algorithm were theoretically derived and the performance parameters of the improved algorithm were analyzed numerically. The analysis results showed that the improved algorithm had better performance in terms of throughput and delay, it had a good reference value for the design of the actual system for future research.

Key words: Ad hoc networking, packet scheduling, modified Galois field design algorithm, throughput, delay.

(C)1994-2020 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net