

基于拥塞控制的无线网络最大权调度算法

陈持鑫,周继鹏

(暨南大学计算机科学系,广州 510632)

摘 要:无线网络中包调度和拥塞控制方案通常是独立设计的,导致网络资源无法高效利用。针对该问题,基于拥塞控制提出一种新的最大权调度算法。根据最大权调度算法计算节点中所有流的权值,选择最大权值的流进行调度,根据网络的拥塞情况,调节节点中流的发送速率。仿真结果表明,该算法能够提高网络吞吐量,达到更好的公平性并减少丢包率。

关键词:无线网络;最大权调度算法;拥塞控制;吞吐量;公平性;丢包率

中文引用格式:陈持鑫,周继鹏. 基于拥塞控制的无线网络最大权调度算法[J]. 计算机工程,2016,42(5):130-133.

英文引用格式:Chen Chixin, Zhou Jipeng. Maximum Weight Scheduling Algorithm for Wireless Network Based on Congestion Control[J]. Computer Engineering, 2016, 42(5):130-133.

Maximum Weight Scheduling Algorithm for Wireless Network Based on Congestion Control

CHEN Chixin, ZHOU Jipeng

(Department of Computer Science, Jinan University, Guangzhou 510632, China)

[Abstract] In wireless network, packet scheduling and congestion control are usually independently designed. This prevents the network resources from being efficiently utilized. In order to solve this problem, a new maximum weight scheduling algorithm based on congestion control is proposed in this paper. The algorithm computes weights of all the flows in the node, chooses the flow with the maximum weight to schedule and adjusts the transmission rate of the flow according to the network congestion situation. Simulation results show that, the algorithm can improve network throughput, achieves better fairness and reduces packet loss rate.

[Key words] wireless network; maximum weight scheduling algorithm; congestion control; throughput; fairness; packet loss rate

DOI:10.3969/j.issn.1000-3428.2016.05.022

1 概述

无线网络是现代通信系统不可或缺的一部分。不像带宽和其他资源丰富的有线网络,无线网络资源是高度受限的,因此,对无线网络资源的有效利用将十分重要。传统的网络协议采用严格的分层结构,包调度和拥塞控制在不同的层独立执行。包调度决定下一个进入网络的数据包。拥塞控制根据网络反馈信息,控制数据流的发送速度。由于无线网络资源的共享性,传统的网络协议不能达到最优的分配速率^[1]。近年来,独立执行包调度和拥塞控制被证明会对网络产生不利影响^[2-4]。联合设计包调

度和拥塞控制对于提高网络吞吐量和有效地利用资源具有重要的意义。文献[5]提出了一种基于拥塞的机会调度算法,减少分组被转发到下一跳节点被丢失的概率,选择质量好的下一跳进行传输,避免节点拥塞所造成的资源浪费问题。文献[6]根据数据包调度是服务质量(Quality of Service, QoS)支持最基本的部分,不合理的数据包调度会造成网络拥塞,而网络拥塞会影响 QoS 支持的效率,提出了一种基于差分队列服务和 Semi-TCP 的联合方案,以提高它们在网络服务质量和拥塞控制方面的性能。文献[7]通过扩展网络效用最大化和应用双基分解框架提出了一种无线 Ad-Hoc 网络联合设计包调度和

基金项目:国家自然科学基金资助项目(61373125,61272073);广东省自然科学基金资助项目(S2013020012865);广东省科技计划基金资助项目(2013B010401016)。

作者简介:陈持鑫(1988-),男,硕士,主研方向为无线传感器网络;周继鹏,教授。

收稿日期:2015-06-29 **修回日期:**2015-08-21 **E-mail:**wallacexin@foxmail.com

拥塞控制的方案。文献[8]提出了一种基于矩阵干扰模型下的包调度和拥塞控制设计方案,使用了2种随机分布式算法,但会给网络带来很大负担。文献[9]提出了无线贪婪对偶原始算法 wGDP (wireless Greedy Duel Primal),这是一种基于无线 Ad-Hoc 网络的联合设计包调度和拥塞控制的算法。wGDP 算法在实现上可以做到对现有的网络协议造成较小冲突,但是它只考虑了网络通信都集中在同一个信道这种最简单的网络模型。文献[10]提出了一种用于非实时流的缓冲感知和流量依赖调度算法,它可以容忍某些包延迟,但对于丢包敏感。传统的最大权调度算法已经被证明在单一基站网络中能达到网络最大吞吐量^[11-13],但是有以下不足:(1)最大权调度假定缓存区队列长度是无限的,这是不现实的。(2)最大权调度是不公平的。(3)最大权调度只适用于基站网络中,在多跳网络中的性能较差。本文在节点队列有限的无线网络中,联合拥塞控制和最大权调度算法,提出基于拥塞控制的调度算法。

2 系统模型

在本文中,一个无线网络可以描述为 $G = \{V, E\}$,其中, V 是节点的集合; E 是无线链路的集合。在这里,有 $|V| = N$ 和 $|E| = L$ 。假设节点采用全向天线,无线链路是双向的。每个节点 v 有一个传输范围 $RT(v)$,这说明只有那些距离节点 v $RT(v)$ 范围内的节点可以正确接收从 v 发送的信号,仅当2个节点在彼此的传输范围内,链路才存在。假设每个节点 v 具有干扰范围 $RI(v)$,在该范围内使用相同信道的其他节点都会受到干扰。为简单起见,假设所有节点都具有相同的传输范围 RT 和干扰范围 RI ,并且每个节点的干扰范围是它们传输范围的2倍,也就是说, $RI = 2RT$ 。在本文的网络模型中,假设所有节点都使用相同的信道进行通信。这种假设是最经常被用来研究标准 802.11 无线协议的。如果在彼此的干扰范围之内,则链路不能同时传送。假设有一组固定的流,它们的发送功率是固定的,路由协议使用 AODV。

每一个节点的调度器结构由3个模块构成:包分类器,缓冲管理器和包调度器(见图1)。包分类器根据包所在的流进行分类,并将它们送入缓冲管理器,不同的流使用不同的队列。缓冲管理器记录不同流的一些相关数据,如到达时间、等待时间、队列长度及空缓冲等。包调度器根据调度优先级选择下一个节点发送的数据包进行发送。传统的最大权调度算法通常假设缓冲空间是无限的,则缓冲区不

会有溢出事件发生,因此,不需要拥塞控制。在本文方案中,使用有限的缓冲空间并且联合考虑拥塞控制。

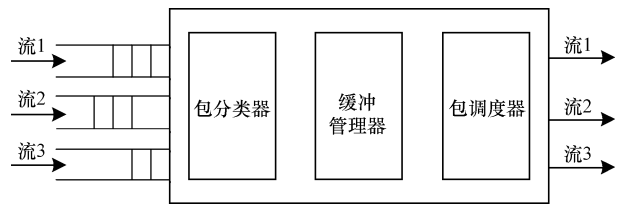


图1 调度器结构

3 基于拥塞控制的调度算法

3.1 设计目的

传统的最大权调度算法一般使用传输的数据包队列长度信息和类型来分配网络资源,以使网络吞吐量最大化^[13-15]。但是它通常假定节点的缓存队列长度是无限的。实际上,传统的最大权调度算法会将资源都分配给队列长度最长的队列,产生不公平的现象。拥塞控制一般和 TCP 紧密联系在一起,是一种端到端的控制策略,它需要不断地交换节点间的信息,这是不容易实现的。调度算法是决定在每个时隙下哪些数据包应该由节点发送。拥塞控制决定着在每个时隙有多少数据包可以被发送到网络中。在目前的互联网上,几乎所有的主机都运行 TCP 协议,而所有的路由器都运行调度算法,但是两者并没有直接交互。因此,本文通过联合拥塞控制和最大权调度算法,提出了基于拥塞控制的调度算法,将拥塞控制在点到点之间实现。基于拥塞控制的调度算法的总体设计目标就是在节点缓冲有限的情况下使得网络具有更好的吞吐量、公平性和丢包率。

3.2 算法设计与描述

基于拥塞控制的调度算法所使用的网络在任意2个节点相互传输数据时,它们都可以交换彼此的队列信息。节点的缓冲管理器可以获得当前队列的信息以及信道的使用状况,同时它也可以获得下一跳节点的队列信息。本文使用 RTS/CTS 的方式来交换节点之间的信息。网络中的任何节点在发送 RTS 和 CTS 包时,将本节点的信息放到 RTS 或 CTS 包中一并发送,而节点在接收到 RTS 或者 CTS 包时,根据目的包的源地址,将有关的信息进行存储。文献[13]已经证明了,基于队列的最大权调度算法是一种稳定的算法。如果流发送的速率被限定在合理的范围内,那么网络的吞吐量将是最优的,因此,节点的队列长度也是稳定的。但是它有一个明显的缺点,那就是它假定所有队列的长度都是无限的,这是不现实的。

本文中的每一个链路 L 中所有的流标记为 $Q_i = \{s_i, i=1, 2, \dots, s_i\}$, 基于拥塞控制的最大权调度算法描述如下, 网络中所有节点在每个调度时间间隔 t 都执行如下算法:

(1) 所有到达节点 u 的包, 根据其所在的流 s_i 存放到对应的队列中, 缓存管理器记录每一个流的缓存大小 $B_u^i(t)$ 和每一个流的当前队列大小 $q_u^i(t)$, $q_u^i(t)$ 初始值为 0, 每当有一个流 i 的包到达时, $q_u^i(t)$ 加 1。

(2) 节点 u 在每个调度时隙都能获得下一跳节点 v 的队列情况, 包括每一个流的缓存大小 $B_v^i(t)$ 和每一个流的当前队列大小 $q_v^i(t)$, 然后计算节点 v 的空缓存率 $BF_v^i(t) = \frac{B_v^i(t) - q_v^i(t)}{B_v^i(t)}$ 。

(3) 在每个调度时隙 t , 缓存管理器计算所有流权重值 $\lceil q_u^i(t) - q_v^i(t) \rceil \times r_i^i(t) \times BF_v^i(t)$ ($r_i^i(t)$ 是 u, v 节点形成链路的数据传输速率), 然后取得权重值最大的队列 $w_i^i(t) = \operatorname{argmax}_{i \in Q_i} \lceil q_u^i(t) - q_v^i(t) \rceil \times r_i^i(t) \times BF_v^i(t)$ 进行发送。若权重值最大的队列有多个, 则取流序号最小的队列进行发送, 同时修改 $q_u^i(t)$ 的值, 令 $q_u^i(t) = q_u^i(t) - r_i^i(t) \times t$ 。

(4) 根据拥塞控制的原理, 调节发送速率, 以实现公平调度。设 $\overline{b_v^i(t)}$ 表示流 i 发送到网络中的包数量, $b_v^i(t)$ 则表示了下一跳节点的接收能力, 比较发送速率与延迟的乘积 $\overline{b_v^i(t)} = \overline{\text{delay}_i} \times r_i^i(t)$ 与下一跳节点的空队列长度 $b_v^i(t) = B_v^i(t) - q_v^i(t)$ 。若 $\overline{b_v^i(t)} < b_v^i(t)$, 则增大发送速率使得 $r_i^i = r_i^i \times (1 + \theta)$; 若 $\overline{b_v^i(t)} > b_v^i(t)$, 则减少发送速率使得 $r_i^i = r_i^i \times (1 - \theta)$ 。其中, θ 是一个调节发送速率的参数, 本文实验取 $\theta = 0.1$ 。

(5) 更新所有的队列。

4 仿真结果与分析

4.1 实验说明

网络仿真器 NS2 是一种面向对象的、离散的事件驱动的开源网络仿真工具, 其组建的主要功能使用 C++ 编写, 运算速度快, 网络参数的配置部分则是使用了 OTcl 编写, 修改方便。本文采用 NS2 作为网络仿真工具, 分别对基于拥塞控制的最大权调度算法和 TCP + 802.11MAC 进行模拟实验, 然后将两者进行比较。实验使用网络流的吞吐量和丢包率作为性能指标。仿真拓扑结构如图 2 所示。每个节点的传输距离是 250 m, 在 1 000 m × 1 000 m 的场景实验。设定所有的数据包大小都为 1 024 Byte, 共用一个信道, 信道容量为 1 Mb/s。实验时间是 150 s。

图 2 所示的拓扑模型是一个线性拓扑结构, 包含 4 个节点, 每个节点距离 200 m, 共 4 条流, 所有的流都是弹性流。流 1 ~ 流 3 都只有一跳, 为短流, 流 4 从节点 1 ~ 节点 4, 共三跳, 为长流。因为相邻节点不能同时传输数据, 所以这些节点每次只能有一个链路可以传输数据。本文通过比较基于拥塞控制的最大权调度算法和 TCP + 802.11MAC 协议 (包调度和拥塞控制分别在 MAC 层和传输层独立实现) 来验证基于拥塞控制的最大权调度算法的性能。

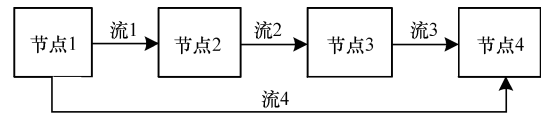


图 2 实验网络拓扑结构

4.2 结果分析

图 3 表示 TCP + 802.11MAC 协议的实验结果, 可以看出, 3 条短流几乎占据了整个网络带宽, 并且它们吞吐量的波动都比较大, 且整体不平滑。而长流几乎在整个模拟时间内得不到调度、吞吐量很小。产生这种现象的原因是在 802.11MAC 随机接入控制下, 当每个节点有数据需要传输时, 都以相同的概率竞争信道, 对于传输跳数越长的包, 竞争信道的次数越多, 获得调度的机会就越少, 因此, 吞吐量越低。

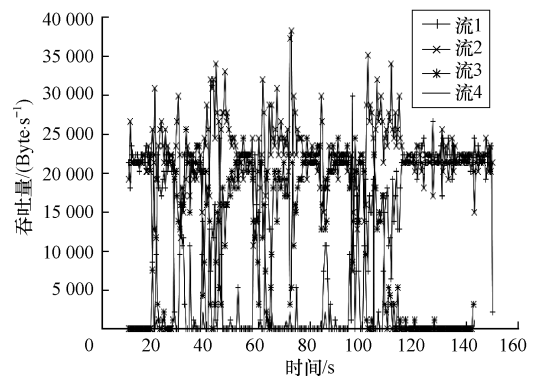


图 3 TCP + 802.11MAC 协议的网络吞吐量

图 4 表示基于拥塞控制的最大权调度算法的实验结果, 可以看出, 3 条短流和长流的吞吐量都较理想。所有流的吞吐量稳定, 波动比 TCP + 802.11MAC 小。并且长流得到充分调度, 体现了调度算法的公平性。整体的网络吞吐量明显高于 TCP + 802.11MAC。产生这种现象的原因是最大权调度算法能使用队列的长度和发送速率等因素进行权值的计算, 选择权值最大的队列进行调度, 达到吞吐量的最优化, 结合拥塞控制策略, 根据网络的发包情况和节点的接收能力, 对发送节点中各个流的发送速率进行调节, 这有利于流的公平性, 使得发送路径较长的流有更多的机会得到调度。

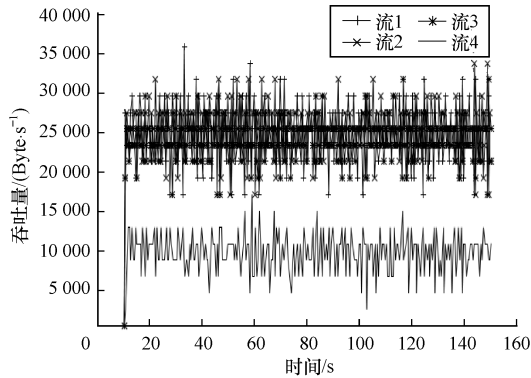


图 4 基于拥塞控制的最大权调度算法的网络吞吐量

图 5 表示 TCP + 802.11MAC 和基于拥塞控制的最大权调度算法的丢包率,可以看出,基于拥塞控制的最大权调度算法能根据权重对队列进行处理,对传输速率高的队列的数据处理能力更强,减少丢包率。实验结果表明其在实验期间的丢包率明显小于 TCP + 802.11MAC。

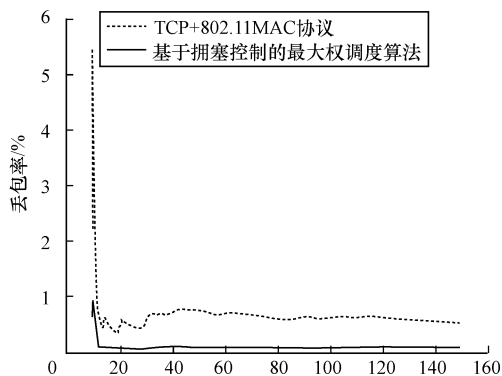


图 5 TCP + 802.11MAC 协议和基于拥塞控制的最大权调度算法的丢包率比较

综上所述,基于拥塞控制的最大权调度算法在吞吐量、公平性和丢包率方面都优于 TCP + 802.11MAC。

5 结束语

本文提出一种基于拥塞控制的最大权调度算法。该算法建立在队列缓冲有限的基础上,公平地对待所有的流。通过在 NS2 上的仿真实验表明,该算法能够有效地提高网络吞吐量,使网络中的流具有更好的平滑性,保障了所有流的公平性,并减少了丢包率。下一步将研究扩展到多信道的无线网络环境下,并结合延迟研究最大权调度算法。

参考文献

[1] Fan Qiu, Yuan Xue. Robust Joint Congestion Control and Scheduling for Time-varying Multi-hop Wireless Networks With Feedback Delay [J]. IEEE Transactions on Wireless Communications, 2012, 13 (9) : 407-415.
[2] Atilla E, Srikant R. Fair Resource Allocation in Wireless Networks Using Queue-length Based Scheduling and

Congestion Control [J]. IEEE/ACM Transactions on Networking, 2007, 15 (6) : 1333-1344.

- [3] Michael J N, Modiano E, Li Chih-Ping. Fairness and Optimal Stochastic Control for Heterogeneous Networks [J]. IEEE/ACM Transactions on Networking, 2008, 16 (2) : 396-409.
[4] Lin Xiaojun, Ness B S. The Impact of Imperfect Scheduling on Crosslayer Rate Control in Multihop Wireless Networks [J]. IEEE/ACM Transactions on Networking, 2006, 14 (2) : 302-315.
[5] 夏海轮, 丁 炜. Ad Hoc 网络中基于拥塞的机会调度算法 [J]. 电子科技大学学报, 2008, 37 (5) : 761-764.
[6] 毛华超. 多跳无线网络服务质量与拥塞控制的联合设计 [D]. 广州: 华南理工大学, 2013.
[7] Chen Lijun, Low S H, Chiang M, et al. Doyle. Cross-layer Congestion Control, Routing and Scheduling Design in Ad Hoc Wireless Networks [C] // Proceedings of the 25th IEEE International Conference on Computer Communications. Washington D. C., USA: IEEE Press, 2006: 1-13.
[8] Gauray S, Ness B S, Ravi R M. Joint Congestion Control and Distributed Scheduling for Throughput Guarantees in Wireless Networks [J]. ACM Transactions on Modeling and Computer Simulation, 2010, 21 (1) : 1-25.
[9] Umut A, Matthew A, Piyush G. Joint Scheduling and Congestion Control in Mobile Ad Hoc Networks [C] // Proceedings of the 27th IEEE Conference on Computer Communications. Washington D. C., USA: IEEE Press, 2008: 1292-1300.
[10] Huang Jinri, Niu Zhisheng. Buffer-aware and Traffic-dependent Packet Scheduling in Wireless OFDM Networks [C] // Proceedings of IEEE Wireless Communications and Networking Conference. Washington D. C., USA: IEEE Press, 2007: 1554-1558.
[11] Matthew A, Zhang Lisa. Scheduling Algorithms for Multicarrier Wireless Data Systems [C] // Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking. New York, USA: ACM Press, 2007: 3-14.
[12] Shreeshankar B, Asnjay S, Lei Ying. Scheduling in Multichannel Wireless Networks: Rate Function Optimality in the Small-buffer Regime [C] // Proceedings of the 11th International Joint Conference on Measurement and Modeling of Computer Systems. New York, USA: ACM Press, 2009: 121-132.
[13] Leandros T, Anthony E. Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks [J]. IEEE Transactions on Automatic Control, 1992, 37 (12) : 1936-1948.
[14] Leandros T, Anthony E. Dynamic Server Allocation to Parallel Queues with Randomly Varying Connectivity [J]. IEEE Transactions on Information Theory, 1993, 39 (2) : 466-478.
[15] Baruch A, Tom L. A Simple Local-control Approximation Algorithm for Multicommodity Flow [C] // Proceedings of the 34th Annual Symposium on Foundations of Computer Science. Washington D. C., USA: IEEE Press, 1993: 459-468.