

A Survey on Congestion Control and Maximization of Throughput in Wireless Networks

K.Ravindra Babu
Post graduate student
V.R.Siddhartha Engineering College
ravindra.bec2008@gmail.com

J.Ranga Rao
Assistant Professor
V.R.Siddhartha Engineering College
rangaraoj@gmail.com

Abstract

In multihop wireless networks, designing distributed Scheduling algorithms to achieve the maximal throughput is a challenging problem because of the complex interference constraints among different links. Traditional maximal-weight scheduling (MWS), although throughput-optimal, is difficult to implement in distributed networks. On the other hand, a distributed greedy protocol similar to IEEE 802.11 does not guarantee the maximal throughput. This proposed system introduces an adaptive carrier sense multiple access (CSMA) scheduling algorithm that can achieve the maximal throughput distributively. Some of the major advantages of the algorithm are that it applies to a very general interference model and that it is simple, distributed, and asynchronous. We are also including congestion control algorithms to reduce the packet loss in the network which improves the packet delivery ratio substantially.

Keywords- Multihop wireless networks, CSMA, Distributed network, Congestion control, Conflict graph, MAC Layer.

1. INTRODUCTION

In Multihop wireless networks, it is important to efficiently utilize the network resources and provide fairness to competing data flows. These objectives require the cooperation of different network layers. The transport layer needs to inject the right amount of traffic into the network based on the congestion level, and the MAC layer needs to serve the traffic efficiently to achieve high throughput. The problem can be naturally decomposed into congestion control

at the transport layer and scheduling at the MAC layer. It turns out that MAC-layer scheduling is the bottleneck of the problem [1]. In particular, it is not easy to achieve the maximal throughput through distributed scheduling, which in turn prevents full utilization of the wireless network. Scheduling is challenging since the conflicting relationships between different links can be complicated. It is well known that maximal-weight scheduling (MWS) [12] is *throughput-optimal*. That is, that scheduling can support any incoming rates within the capacity region. In MWS, time is assumed to be slotted. In each slot, a set of nonconflicting links that have the maximal weight are scheduled, where the "weight" of a set of links is the summation of their queue lengths. Its distributed implementation is not trivial in wireless networks.

In this paper, a distributed adaptive CSMA algorithm is used under the idealized conditions. It is throughput-optimal in wireless networks with a general interference model. We have utilized the product-form stationary distribution of CSMA networks in order to obtain the distributed algorithm and the maximal throughput.

Furthermore, we have combined that algorithm with congestion control to approach the maximal utility and showed the connection with back-pressure scheduling. The algorithm is easy to implement, and the simulation results are encouraging.

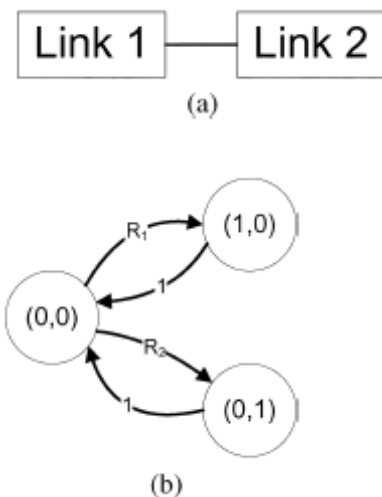


Fig.1. Example: A conflict graph and corresponding CSMA Markov chain. (a) Conflict graph. (b) CSMA Markov chain.

2. RELATED WORK

A few recent works proposed throughput-optimal algorithms for certain interference models. For example, Eryilmaz *et al.* [3] proposed a polynomial-complexity algorithm for the "two-hop interference model".¹ Modiano *et al.* [4] introduced a gossip algorithm for the "node-exclusive model".² The extensions to more general interference models, as discussed in [3] and [4], involve extra challenges. Sanghavi *et al.* [5] introduced an algorithm that can approach the throughput capacity (with increasing overhead) for the node-exclusive model. On the other hand, a number of low-complexity but suboptimal scheduling algorithms have been proposed in the literature. By using a distributed greedy protocol similar to IEEE 802.11, [8] shows that only a fraction of the throughput region can be achieved (after ignoring collisions).

The fraction depends on the network topology and interference relationships. The algorithm is related to Maximal Scheduling [9], which chooses a maximal schedule among the nonempty queues in each slot. Different from Maximal Scheduling, the Longest-Queue-First (LQF) algorithm [10]-[13] takes into account the queue lengths of the nonempty queues.

2.1 Maximal-weight scheduling (MWS)

Maximal-weight scheduling (MWS) [12] is *throughput-optimal*. That is, that scheduling can support any incoming rates within the capacity region. In MWS, time is assumed to be slotted. In each slot, a set of non-conflicting links (called an "independent set," or "IS") that have the maximal weight are scheduled, where the "weight" of a set of links is the summation of their queue lengths. (This algorithm has also been applied to achieve 100% throughput in input-queued switches [13].) However, finding such a maximal-weighted IS is NP-complete in general and is hard even for centralized algorithms. Therefore, its distributed implementation is not trivial in wireless networks.

2.2 Longest-Queue-First (LQF) algorithm

The Longest Queue First (LQF) algorithm [10]-[13] takes into account the queue lengths of the nonempty queues. It shows good throughput performance in simulations. In fact, LQF is proven to be throughput-optimal if the network topology satisfies a "local pooling" condition [10], [12] or if the network is small [13]. In general topologies, however, LQF is not throughput-optimal, and the achievable fraction of the capacity region can be characterized as in [11]. Reference [14] studied the impact of such imperfect scheduling on utility maximization in wireless networks. In [13], Proutiere *et al.* developed asynchronous random-access-based scheduling algorithms that can achieve throughput performance similar to that of the Maximum Size scheduling algorithm. In general topologies, however, LQF is not throughput-optimal, it is suitable to small networks. It will not give optimal throughput in large networks.

3. CSMA Algorithm

Distributed adaptive carrier sense multiple access (CSMA) algorithm for a general interference model. It is inspired by CSMA, but may be applied to more general resource sharing problems (i.e., not limited to wireless networks). If the packet collisions are ignored the algorithm can achieve maximal throughput. When more than one station attempts to transmit a frame at the same time, a *collision* occurs, and subsequently all frames get corrupted. The standard mechanism for contention resolution in computer networks is called *carrier-sense multiple access* (CSMA). CSMA algorithms attempt to break symmetries of

failing transmissions being restarted at almost the same time by using randomized binary exponential backoff procedures. While wired devices can listen during their own transmissions and employ CSMA with collision detection (CSMA/CD), stations in wireless networks usually cannot listen to their own transmissions, and consequently colliding transmissions can only be detected after they have been completed. Thus wireless devices use CSMA with collision avoidance (CSMA/CA or CSMA-CA).

The CSMA-CA algorithm works as follows: When a station or device or node wants to send some information, the algorithm works as follows

Main procedure of csma algorithm

- I. Is my frame ready for transmission? If yes, it goes on to the next point.
- II. Is medium idle? If not, wait until it becomes ready
- III. Start transmitting.
- IV. Did a collision occur? If so, go to collision detected procedure.
- V. Reset retransmission counters and end frame transmission.

Collision detected procedure

1. Continue transmission until minimum packet time is reached to ensure that all receivers detect the collision.
2. Increment retransmission counter.
3. Was the maximum number of transmission attempts reached? If so, abort transmission.
4. Calculate and wait random back off period based on number of collisions.
5. Re-enter main procedure at stage 1.

CSMA Distinct features

Check the next node before send the data. Each node only uses its local information (e.g., its backlog). No explicit control messages are required among the nodes. It is based on CSMA random access, which is similar to the IEEE 802.11 protocol and is easy to implement. Time is not divided into synchronous slots. Thus, no synchronization of transmissions is needed.

Cross Layer Congestion Control Technique

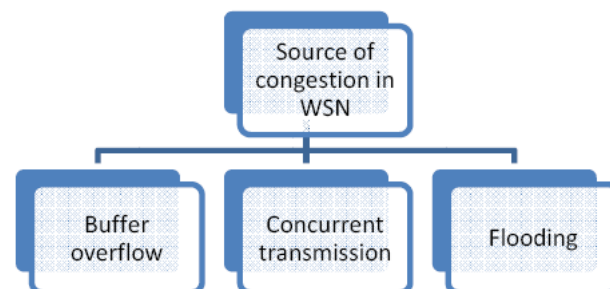


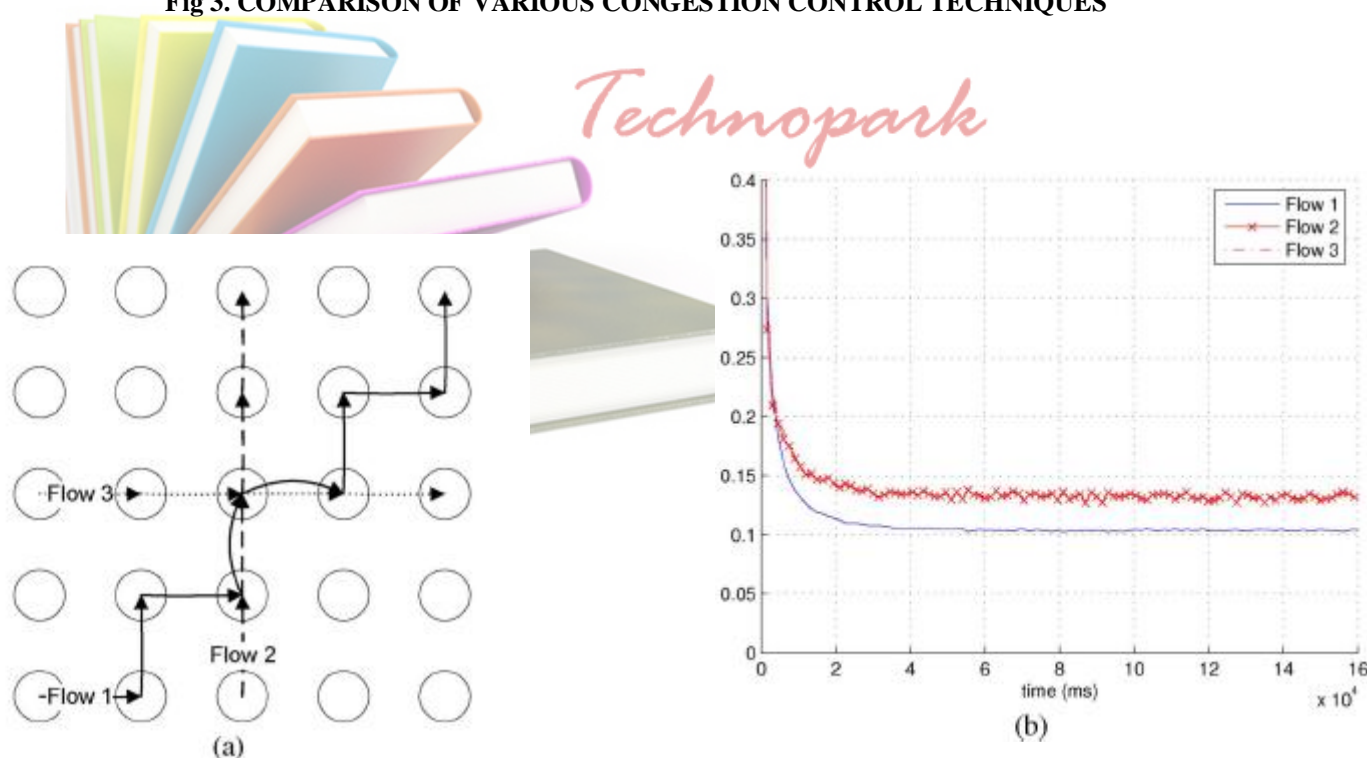
Fig.2. Various Sources of Congestion

Cross layer congestion control technique used for congestion control in the proposed paper which can achieve maximum throughput in wireless networks. Congestion in a network is characterized by delay and packet loss in the network. Transport Control Protocol (TCP) is used as a reliable transport layer protocol in the traditional best effort (wired) network and deals with congestion effectively. The congestion control mechanism of various versions of TCP provides better throughput in wired network, where the packet loss is mainly due to congestion at various nodes and routers. However, this mechanism may not be suitable in a wireless network, where packet loss due to time-varying nature of channel and interference of other nodes are considerably high. Hence, instead of usual congestion control technique, we propose a cross layer technique involving TCP and MAC (Medium Access Control) layer. TCP layer performs the windowing flow control and MAC layer varies transmission power of wireless nodes depending on the channel condition and interference. Our approach consists of:

- 1) Formulation of TCP congestion control mechanism in terms of control system equations.
- 2) Use of transmission power of wireless nodes as a function of cost in an optimization equation.
- 3) Use of optimization techniques to determine the maximum aggregate utility of all the sources, subject to capacity constraints and maximum transmission power of wireless nodes.

Technique	Congestion Detection	Congestion Control	Action Initiator	Application specific Importance
Locality Driven Congestion control	Not available	BY assigning priority to each event	All data sensing node	Self organizing WSN i.e. Weather sensor network
Cluster based congestion control	Traffic Intensity Approximation Model	Rate limiting	Source node	Implemented for large area
Mitigating Congestion in Wireless Sensor Networks	Channel sampling	Hope-by –Hope flow control	Any node	Note Specified
Intelligent packet dropping algorithm	occupancy of the buffer $CD=PA/PB$	assigning priority to the data packets	Intemidiate node	In flooding environment where data packet having limited importance
Event to Sink Reliable Transport	By set ting Congestion notification bit in the ACK packet	By choosing the unique path each event	Sink node	Reliable event detection

Fig 3. COMPARISON OF VARIOUS CONGESTION CONTROL TECHNIQUES



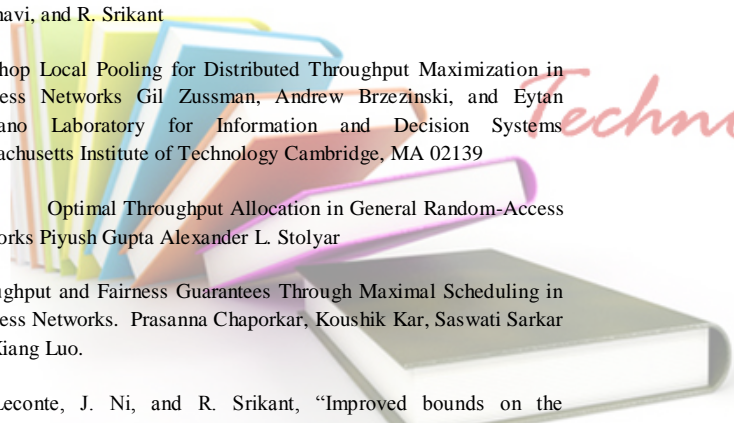
Flow Rates

Fig.4 Flow rates in Network 2 (Grid Topology) with joint scheduling and congestion control (a) Network 2 and flow directions (b) Flow rates

4. Conclusion and Future Work

In this paper, a distributed adaptive CSMA algorithm is used under the idealized conditions. It is throughput-optimal in wireless networks with a general interference model. We have utilized the product-form stationary distribution of CSMA networks in order to obtain the distributed algorithm and the maximal throughput. Furthermore, we have combined that algorithm with congestion control to approach the maximal utility and showed the connection with back-pressure scheduling. The algorithm is easy to implement, and the simulation results are encouraging. Here the packet delivery ratio is very high when compared to previous techniques but still one can improve the packet delivery ratio by using some advanced congestion control techniques.

5. REFERENCES

- 
- [1] Distributed Link Scheduling with Constant Overhead Loc Bui, Sujay Sanghavi, and R. Srikant
- [2] Multihop Local Pooling for Distributed Throughput Maximization in Wireless Networks Gil Zussman, Andrew Brzezinski, and Eytan Modiano Laboratory for Information and Decision Systems Massachusetts Institute of Technology Cambridge, MA 02139
- [3] Optimal Throughput Allocation in General Random-Access Networks Piyush Gupta Alexander L. Stolyar
- [4] Throughput and Fairness Guarantees Through Maximal Scheduling in Wireless Networks. Prasanna Chaporkar, Koushik Kar, Saswati Sarkar and Xiang Luo.
- [5] M. Leconte, J. Ni, and R. Srikant, "Improved bounds on the throughput efficiency of greedy maximal scheduling in wireless networks," in *Proc. ACM MobiHoc*, May 2009, pp. 165–174.
- [6] X. Lin and N. Shroff, "The impact of imperfect scheduling on crosslayer rate control in multihop wireless networks," in *Proc. IEEE INFOCOM*, Miami, Florida, Mar. 2005, vol. 3, pp. 1804–1814.
- [7] P. Marbach, A. Eryilmaz, and A. Ozdaglar, "Achievable rate region of CSMA schedulers in wireless networks with primary interference constraints," in *Proc. IEEE Conf. Decision Control*, 2007, pp. 1156–1161.
- [8] Proutiere, Y. Yi, and M. Chiang, "Throughput of random access without message passing," in *Proc. Conf. Inf. Sci. Syst.*, Princeton, NJ, Mar. 2008, pp. 509–514.
- [9] A. Dimakis and J. Walrand, "Sufficient conditions for stability of longest-queue-first scheduling: Second-order properties using fluid limits," *Adv. Appl. Probab.*, vol. 38, no. 2, pp. 505–521, 2006.
- [10] C. Joo, X. Lin, and N. Shroff, "Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks," in *Proc. IEEE INFOCOM*, Phoenix, AZ, Apr. 2008, pp. 1103–1111.
- [11] G. Zussman, A. Brzezinski, and E. Modiano, "Multihop local pooling for distributed throughput maximization in wireless networks," in *Proc. IEEE INFOCOM*, Phoenix, AZ, Apr. 2008, pp. 1139–1147.
- [12] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1234–1241, 1992.
- [13] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," *IEEE Trans. Commun.*, vol. 47, no. 8, pp. 1260–1267, Aug. 1999.