Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 9, September 2014, pg.764 – 770

RESEARCH ARTICLE

DISTRIBUTED SINR-BASED SCHEDULING ALGORITHM FOR MULTIHOP WIRELESS NETWORKS

Miss. S.Vijayalakshmi[1], Mrs. R.Kavitha[2]
[1]M.Phil Research Scholar, Department Of Computer Science, Vivekanandha College for Women, Namakkal (India)
[2]Assistant Professor, Department Of Computer Science, Vivekanandha College for Women, Tiruchengode, Namakkal (India)
[1] laxviji230611@gmail.com, [2] kavithamscmphil@gmail.com

ABSTRACT: The problem of developing distributed scheduling algorithms for high throughput in multi hop wireless networks has been extensively studied in recent years. The design of a distributed low-complexity scheduling algorithm becomes even more challenging when taking into account a physical interference model, which requires the SINR at a receiver to be checked when making scheduling decisions. To do so, we need to check whether a transmission failure is caused by interference due to simultaneous transmissions from distant nodes. In this paper, we propose a scheduling algorithm under a physical interference model, which is amenable to Distributed implementation with 802.11 CSMA technologies. The proposed scheduling algorithm is shown to achieve throughput optimality. We present two variations of the algorithm to enhance the delay performance and to reduce the control overhead, respectively, while retaining throughput optimality.

Keywords: Wireless scheduling, SINR, CSMA, discrete time Markov chain

I. INTRODUCTION

IT is widely acknowledged that link scheduling is a major performance bottleneck in wireless multihop networks. Link scheduling determines which transmitter-receiver pairs (links) are to be simultaneous activated at a given moment in order to achieve high throughput, low delay, fairness, etc. Over the past couple of decades, a variety of link scheduling algorithms under different interference models have been studied in order to achieve high performance and low complexity. In wireless multihop communication environments, simultaneous link activation could cause significant mutual interference that results in transmission failure if, e.g., the interference exceeds a certain threshold. Also, for a practical implementation, it is desirable to design distributed algorithms for link scheduling. Therefore, developing high-performance distributed scheduling solutions that are amenable to practical implementation is one of the most significant challenges in multihop wireless networks. So far, many scheduling schemes, centralized or distributed, have been studied in the literature under different interference models with different time granularities (continuous or time-slotted). The problem of achieving throughput optimality in wireless multihop networks has been extensively studied. The well-known Max-Weight Scheduling (MWS) algorithm has been shown to achieve throughput optimality at the cost of the very high time complexity. In a time slotted system, the MWS algorithm picks, at each time slot, the set of nonconflicting (e.g., not causing transmission failures) links whose queue-weighted sum is the largest. In general, finding such a set of max-weighted nonconflicting links is NP-hard and requires centralized implementation. Many suboptimal (and hence more practical) solutions have been proposed over the past several years aimed to reduce this algorithmic complexity.

II. NETWORK MODEL

A wireless network is modeled by a graph G (V, E), where V denotes the set of nodes and E denotes the set of links. We represent a directed link by an ordered pair of nodes. Two nodes are directly connected by a link if they can successfully exchange packets when there is no other transmission. We assume that the transmission power P is Fixed, and that a single frequency channel is available for the whole system; that is, two or more links that transmit at the same time may interfere with each other. The transmission of a packet over a link will be successful if the SINR at the receiver is above a certain SINR threshold. We assume that a single SINR threshold is used for all the links. The transmission rate and its corresponding SINR threshold can be carefully chosen by considering the network density. All links are assumed to have unit capacity (i.e., a packet can be transmitted at a unit time) and we consider a time slotted System, where each time slot consists of a control slot and a data slot. A transmission failure occurring the SINR at the receiving node of a link is below a certain threshold. The control slot is intended to determine a “feasible” transmission schedule for the data slot. A feasible schedule means that the links in the schedule can be activated without causing transmission failures. The data slot is used for the scheduled links to transmit a data packet. A feasible schedule is defined as a set of links that can be active simultaneously, and each link has a sufficient SINR value at the receiver. We assume that data packets arrive at a link and leave the system immediately once it is transmitted over the link. Although we consider only single-hop traffic, scheduling difficulty lies in multihop nature of wireless interference. The capacity region of a scheduling policy is the set of arrival rates for which there exists a scheduling algorithm that can stabilize the network. The network is said to be stable if the expected queue lengths of all links remain finite. Let M be the set of all feasible schedules in our network model.

III. DISTRIBUTED SINR-BASED SCHEDULING ALGORITHM

In this section, we describe a basic throughput-optimal scheduling scheme that operates in a distributed manner under the SINR interference model specified in the previous section. We describe this scheduling algorithm called DSS and analyze its throughput performance.

3.1 Algorithm Description

At each time slot t, we decide a transmission schedule by reusing some of the activated links from the previous transmission schedule at slot, and adding new links. To add new links, let m (t) denotes a random candidate vector at time slot t, which refers to a set of randomly selected links. From m (t), we derive a new “feasible” addition vector d(t) using the procedure described in the next paragraph. Then, for each link 2 d(t), the proposed Scheduling algorithm decides probabilistically whether will be activated or not. The difficult part of the procedure is to find a feasible addition vector d(t) in a distributed manner. To this end, we assume that a receiver can differentiate 1) the received signal strength (RSS) if the signal is transmitted from its sender, and 2) the interference power if the signal is Transmitted from an interferer. This can be calculated from a premeasured radio frequency (RF) profile. In the following, we describe the proposed scheduling algorithm with a focus on how to find d(t).

IV. DSS-P

We have improved the delay performance of CSMA-based scheduling using the dual-state approach. Even though we have reduced the total queue length of the network, distributed CSMA-based scheduling schemes have another drawback: the overhead incurred due to the control mini slots. As the traffic load increases, the performance of CSMA-based approaches becomes better as long as there are a sufficient number of control mini slots. This overhead can be significant. Let us take the IEEE 802.11a OFDM PHY for example; the length of a single back off slot. Then, the length of a single control mini slot of DSS-family scheduling algorithms would be at least since each control mini slot has two phases. Since it takes 2 ms to transmit a 1,500 byte long packet at 6 Mbps transmission rate (excluding the PHY/MAC header overhead for simplicity), the time overhead for the control slot exceeds the data transmission time if the number of mini slots M is larger than 111. This control overhead becomes worse as the length of the data packet decreases and/or the transmission rate increases. Due to the random nature of the back-off process, the more links need to be scheduled, the more control mini slots are required. To reduce the control mini slot overhead, we propose DSS-P which replaces the random back-off process by the p-persistent CSMA contention mechanism. In DSS-P, each sender in m(t), attempts to transmit a control packet with the link activation probability p for each mini slot. (The link activation probability p is the same as the link activation probability.) Then, we can check whether the attempting links constitute a feasible schedule by making the receivers calculate the SINR threshold as before. Whenever a feasible link set is obtained, there are no busy tones and a feasible schedule is found.

4.1 Throughput Optimality of DSS-P

In this section, we briefly prove how the DSS-P algorithm can achieve throughput optimality. We first show that the final set of activated links at slot t, x (t), can evolve as a DTMC just like the DSS algorithm. Next, we make a product-form stationary distribution for the transition probability between two states. Then, we will show that throughput optimality can be satisfied. Let M0 (m(t)) (a subset of M) denote the set of all the feasible schedules that can be probabilistically derived from a given random candidate vector m(t). Since the transmission schedule x(t) only depends on the previous transmission schedule x(t-1) and the random candidate vector m(t), we can model x(t) as the state of a DTMC. Then, we need to derive the transition probability between two transmission schedules (or states) x (t- 1) and x (t)under DSS-P. According to the DSS-P algorithm, the feasible link set is probabilistically determined by the link activation probability p at a certain mini slot. Then, the determined schedule is directly used as the final transmission schedule x (t).

4.2 Extension of DSS-P

Even though DSS-P achieves throughput optimality, we can enhance its performance by opportunistically finding a Schedule with a greater number activated links. In the original DSS-P, the process to find a feasible schedule is over when a feasible link set is obtained for the first time even if there are still remaining control mini slots. If these remaining control mini slots can be exploited to find a better schedule, then the delay performance can be further improved. To this end, we continue the probabilistic attempt process to find a better schedule incrementally until the end of the control mini slots. That is, even if we have already obtained a feasible schedule, we seek to add some of the remaining links while satisfying the feasibility to the obtained schedule. The extended DSS-P operates as follows: After finding the first feasible link set at a certain mini slot, more links can be added in addition to the feasible schedule at a later mini slot as long as the SINR constraints of the already scheduled links are not violated. Note that the addition of the links to the schedule can take place multiple times. As additional links will cause more interference to the already scheduled links, we should be conservative. That is, the link activation probability, p, may as well be reduced. To increase the chances of adding more links to the schedule, we geometrically reduce the link activation probability (of the other remaining links in m (t)) whenever the feasible link set is augmented. In our extended DSS-P algorithm, whenever more links are added to the schedule, other remaining candidate links reduce the current activation probability by multiplying by p. Since this extension is similar to the dual-state approach, the throughput optimality of the extended DSS-P algorithm can be proven similarly and hence skipped.

V. PERFORMANCE EVALUATION

We evaluate the performance of DSS, DSS-D, and DSS-P with other representative scheduling schemes in the literature under the SINR-based interference model. We consider a network with nodes that are placed on an area of 100 square units. We construct topologies as follows: We first randomly select the position of a sender uniformly in the area, and locate its corresponding receiver at a random place uniformly within 10 units from the sender. Repeating the locating processes, we generate two topologies; one with 49 and the other with 196 links. (That is, they Have total 98 and 392 nodes, respectively.) The signal transmitted by a sender attenuates as it propagates over space. For the radio propagation model, we adopt the simple two-ray ground model and all the other channel effects (e.g., short and long term fading) are not considered. At the receiver, we assume that the signal can be decoded if the SINR is over a certain threshold, and that all the links have the same SINR threshold value, which is set to 10 dB. For each link, we consider single-hop traffic according the Poisson process with (packet) arrival rate being either 0.1 or 0.9 (evenly at random). Note that the traffic load (which is a simulation parameter) should be multiplied to the packet arrival rate for the effective traffic load of each link. Our performance metric corresponding to the measured total queue lengths of all the links and the throughput after 5,000 time slots. For each plot, we obtain the average over 20 simulation runs with different arrival rate patterns. We evaluate the performance of CSMA-based scheduling Schemes including DSS, DSS-D, DSS-H, and a centralized scheme of GMS (which is an omniscient scheduling scheme). As DSS can be considered as Q-CSMA under the SINR-based interference model, DSS-H can be considered as a counterpart of HQ-CSMA under the SINR. Delay Performance of DSS and DSS-D We first evaluate the delay performance of DSS and DSS-D for various values of M, as the traffic load increases. The total queue lengths of DSS with 49 and 196 links, respectively. We set the attempt probability pa to 0.1, with which a link attempts to include itself in m (t). As the number of control mini slots increases, DSS shows lower delay performance in both topologies. However, for both topologies, beyond a certain value of M, the performance gains are of diminishing value.

VI. CONCLUSION

In this paper, we investigate the scheduling problem in multihop wireless networks under realistic SINR-based interference model. We first develop a fully distributed throughput optimal baseline scheme, called DSS that leverages carrier sensing and exploits recent result in throughput optimality. We then extend this scheme in various ways. We improve delay performance by developing DSS-D that separates activation states for data transmission from virtual states for state transition, thus reducing the delay while achieving throughput optimality. We also propose DSS-P to reduce the control overhead. DSS-P replaces a random back-off process of DSS (or DSS-D) with p-persistent CSMA. We show that DSS-P is throughput optimal and uses far less overhead, outperforming the other CSMA-based schemes for a large class of topologies. There are many open problems in scheduling under the SINR model. Since most recent communication technologies allow rate adaptation and/or variable packet sizes based on the received SINR level, it is interesting to develop CSMA based schemes that achieve high performance under time varying link rates. Understanding transitions of Markov Chain state and achieving convergence of state distribution are of particular interest. Also, getting timely SINR feedback from receivers will increase overhead and requires further research to reduce the complexity.

REFERENCES

[1] A. Dimakis and J. Walrand, "Sufficient Conditions for Stability of Longest-Queue-First Scheduling: Second-Order Properties Using Fluid Limits," Advances in Applied Probability, vol. 38, no. 2, pp. 505-521, 2006.

[2] J.-H. Hoepman, "Simple Distributed Weighted Matchings," http://arxiv.org/abs/cs/0410047v1, Oct. 2004.

[3] C. Joo, "A Local Greedy Scheduling Scheme with Provable Performance Guarantee," Proc. ACM MobiHoc, 2008.

[4] L. Jiang and J. Walrand, "A Distributed CSMA Algorithm for Throughput and Utility Maximization in Wireless Networks," Proc. Allerton Conf., 2008.

[5] P. Marbach and A. Eryilmaz, "A Backlog-Based CSMA Mechanism to Achieve Fairness and Throughput-Optimality in Wireless Networks," Proc. Allerton Conf., 2008.

[6] J. Ni and R. Srikant, "Distributed CSMA/CA Algorithms for Achieving Maximum Throughput in Wireless Networks," Proc. Information Theory and Applications Workshop (ITA), 2009.

[7] J. Ni, B. Tan, and R. Srikant, "Q-CSMA: Queue-Length Based CSMA/CA Algorithms for Achieving Maximum Throughput and Low Delay in Wireless Networks," Proc. IEEE INFOCOM, 2010.