Shadow Queue Based Fair Scheduling in Multi-hop Wireless Networks

XU Jian
College of Computer
Hangzhou Dianzi University
Hangzhou, China
e-mail: jian.xu@hdu.edu.cn

GUO Ming
City College
Zhejiang University
Hangzhou, China
e-mail: guomail@myrealbox.com

XU Ming
College of Computer
Hangzhou Dianzi University
Hangzhou, China
e-mail: mxu@hdu.edu.cn

LIN Xin
School of Information Science and Technology
East China Normal University
Shanghai, China
e-mail: xlin@cs.ecnu.edu.cn

***Abstract*— we study the problem of flow based scheduling in multi-hop wireless networks. And propose a Max Flows First-Shadow queue based Congestion Control (MFF-SCC) algorithm for combined scheduling and congestion control that aims to solve max-min fairness problem. We give out detailed analysis of the algorithm and also conduct simulations to verify its performance. The results of simulations indicate that the proposed approach can improve fairness of scheduling.**

Keywords- Multi-hop;Shadow Queue;Scheduling

I. INTRODUCTION

In this paper we consider the max-min fairness problem of scheduling algorithms for multi-hop wireless networks. During packets flow from source to destination, they will traverse multiple intermediate nodes in the network. For the shared nature of wireless channel, nodes within radio range of each other will compete for resources. Since there are many nodes, it is important to allocate these shared resources in a fair manner among different nodes. Differ from wired network, the time-varying nature of the wireless environment, coupled with different channel conditions for different nodes, brings significant challenge to get these goals. Moreover, the lack of arrival statistics further complicates the solution. In the past few years, efficient and fair scheduling algorithms in multi-hop wireless networks have been the subject of intensive and interesting research [2][4][5].

Fairness of service and effective of system are main goals of resources allocation in wireless network. One particular allocation algorithm called the Back Pressure algorithm was shown to be throughput-optimal [12] and it can be combined with congestion control to proportion fairly allocate resources among competing nodes in a wireless network. Back-pressure algorithm requires the queue length of every flow that passes through a node to perform resource allocation. It schedules the link flow with the largest differential backlog between adjacent nodes. For example, a flow $f(i)$, pass through adjacent node n、m, when equation (1) denotes differential backlogs between adjacent nodes, then $S^*[t]=(S_1[t],\ldots,S_n[t])$ in equation (2) is the scheduling vector according to Back Pressure algorithm, where Q denote the queue length in different nodes.

$$w_{f(i)}[t]=(Q^{n}_{f(i)}[t]-Q^{m}_{f(i)}[t]) \qquad (1)$$

$$S^*[t]=\max_{S\in\Gamma}\sum_{(n,m)} s_{f(i)}w_{f(i)}[t] \qquad (2)$$

But as mentioned in [3], it is known that these two objectives may conflict with each other. Maximizing the total system throughput may lead to serious unfairness problems, e.g., some flows may be starved. And maintaining strict fairness may significantly reduce system efficiency, as the example showed in Figure 1.

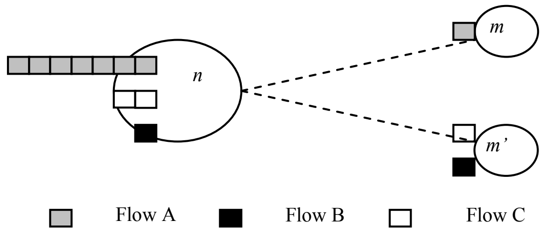

Figure 1. Faireness of Back Pressure algorithm

There are three flows named flow A、B and C separately, their source is on node n and destinations are node m and m'. The destination of flow B and C is node m'. Obviously, when $w_A=6$, $w_B=1$ and $w_C=0$ in equation (1). According to Back Pressure algorithm, link nm will acquire more chances than nm' in node n, then flow B and C will attain less bandwidth. Finally user A、B and C are treated unfairly concerning transmission time slot in this scenario. Due to the fairness problem of throughput first algorithm, extensive researches are done to deal with fair scheduling in multi-hop wireless network. But most works are based on existing method and combine with utility functions. In other word, they all need data source to adapt their sending rate according the function to achieve proportion fairness. And they also can't be

Supported by ZJNSF Grant Y106427.

implemented in practice without disruption to existing wireless protocols. Related works will be outlined in section 5.

Realizing the problem of existing scheduling algorithm, we try to improve the performance of scheduling from a different perspective. We propose a link schedule method called MFF(Max Flows First), which will choose the link to transmission with more flows; and an congestion control method based on shadow queue length of each flow, which named SCC (Shadow queue based Congestion Control). The goal of MFF is to achieve fairness among links and the target of SCC is fairness among different flows in same link. We name these two methods as MFF-SCC algorithm. MFF-SCC keeps separate queues for different links and separate shadow queue for different flows. As show in following analysis, the implementation of MFF-SCC is simple; however its performance is good. In this paper we address such issues. In particular:

①Differ from traditional max-min definition, we use a time slot based max-min fairness definition in corresponding to features of wireless network. And propose MFF-SCC algorithm.

②Fairness of the algorithm is also analyzed.

③ We show via ns2 simulation that MFF-SCC significantly outperforms Back Pressure algorithm operating in conjunction with TCP.

The paper is organized as follows. We describe the formalization of system model and the detailed MFF-SCC algorithm in Section II. Then we discuss the fairness of the algorithm in section III. In Section IV, the simulation results are described. In Section V, we outline the related works. And conclude in Section VI.

II. SYSTEM MODEL AND ALGORITHM

A. SYSTEM MODEL

Let us consider a network modeled by a directed graph, $G=(V, L)$, where V is the set of nodes and L is the set of links. A link $(j, j+1) \in L$ only if node $j+1$ is in the transmission range of node j. It is possible for j to transmit packets from node j to node $j+1$ subject to the interference constraints. Let F be the set of N flows that share resources of the network G. Packets of each flow enter the network G at one node, traverses multiple intermediate nodes (which may or may not pre-determined), and then exit the network at another node. That is, for flow $f_i=(v_0^i, v_{|P_i|}^i) \in F$, let v_0^i denote the entering (begin) node, and $v_{|P_i|}^i$ denote the exiting (end) node of flow f_i. Network G always runs standard routing protocol link AODV. In such a network, a flow f_i always travel along the same path $P_i = \langle v_0^i, v_1^i, \ldots v_j^i, v_{j+1}^i, \ldots, v_{|P_i|}^i \rangle$, where v_0^i denote the entering node, and $v_{|P_i|}^i$ denote the exiting node and v_j^i is a node with distance j to v_0.

The network is time synchronized, and that the transmitter entering node of each flow is associated with an external and independent arrival process $A_i(t)$, which is the cumulative number of packet arrivals during the first time slots. Let $A_i(t)$ denote the number of packet belongs to flow i entering network in time slot t on node v_0. Let vector $A(t)=(A_1(t),\ldots,A_N(t))$ denote the arrival of N flows, and vector $\lambda=(\lambda_1,\ldots,\lambda_N)$ denote corresponding rate.

At each node, we keep a separate FIFO (first-come first-served) queue for each outgoing link, and a separate shadow queue for each outgoing flow. In another word, we only maintain flow number in each link and packet number q (shadow queue length) of each flow in queue.

Instead of keeping a separate real queue Q for each flow as in the Back Pressure algorithm, packets are stored in FIFO queue for all flows going through the corresponding link. It is obvious that packet number counter is much easier to implement than a physical queue. Then MFF-SCC is easier to implement than other max-min scheduling algorithm.

When a node receives a packet, it checks the head of packet: if the node is the destination of this packet, it will deliver to upper layer protocol; otherwise, is will send this packet to real queue Q of the link.

Let $Q_j^l(t)$ denote queue length of link l on node j, and keep count the number of flow $X_j^l(t)$ in this link. $X_j^l(t)$ is also the number of shadow queue. Let $q_j^{i,l}(t)$ denote the length of shadow queue. $q_j^{i,l}(t)$ is also the number of packets of flow f_i in the beginning of time slot t on node j. If equation $i \in l$ represents flow i is passing through link l, then we have equation (3)

$$Q_j^l(t) = \sum_{i \in l} q_j^{i,l}(t) \tag{3}$$

We consider two-hop interference model in IEEE802.11 protocol. And define a valid schedule to be a set of link rates that can be simultaneously supported. Let a activation vector $S[t]$ to denote such a schedule, in each time slot, then the value of vector $S[t]$ represent schedule under given policy and interference model.

$$S_j^i(t) = \begin{cases} 1 & \text{if } q(t)>0\text{, and flow is sceduled} \\ 0 & \end{cases} \tag{4}$$

If one packet is transmitted in single slot, the shadow queue length in this network is expected to change like

$$q_j^{i,l}(t+1) = q_j^{i,l}(t) - S_j^i(t) + S_{j-1}^i(t) \tag{5}$$

B. ALGORITHM

MFF permits a link with more flows to transmit first. The node computes the time slot in which each link can injects packets into the air according to its flow number. For the example in Figure 1, flow A、B and C will be served under the statistic multiplexing principle of FIFO during idle. But when it's congestion in the network, flow A、B and C all have lots of packets to transmit. Channel allocation will in accordance with the number of flows in each link. For there are two flows in link *nm'*, *nm'* will get double slot time to transmit its packets.

In specialized time slot, let vector $X[t]$ denote flow number of each link

$$\mathrm{X}[t] := (X_1[t], ..., X_n[t]) \qquad (6)$$

Consider following schedule of priorities for each link

$$\mathrm{S}[t] := (\mathrm{S}_1[t], ..., \mathrm{S}_n[t]) \qquad (7)$$

Let Γ be the set of all possible valid schedules, and J be channel condition. We must solve the problem

$$\mathrm{S}[t] \in \arg\max_{S\in\Gamma} \sum_{i=1}^{n} X_i[t]J \qquad (8)$$

SCC does fair channel allocation or congestion control for flows. We set an upper-bound *Max_Q* of the arrival rates for each flow based on the memory size of node. Where *Max_Q* is a positive parameter. It shows maximum packets that can keep in memory for each flow. When shadow queue size satisfy equation (9) on node j during congestion,

$$q_j^i[t] > \mathrm{Max\_}Q \qquad (9)$$

SCC will drop packet or send congestion notification to source in probability as follows

$$\min\{(q_j^i[t] - \mathrm{Max\_}Q)/\mathrm{Max\_}Q, 1\} \quad (10)$$

The source will adjust its sending rate at which it injects packets into the network. Here congestion is indicated by memory overhead on the node.

III. FAIRNESS ANALYSIS OF ALGORITHM

Definition 1 clique [11]: A clique is a subset of a graph G such that for all distinct node pair $j, j+1 \in V'$, has the edge $(j, j+1) \in L'$. Flows transverses clique will compete for resources, in one time slot there is only one flow can transmit its packets.

Definition 2 maximal clique: A maximal clique represents a maximal set of those sub-flows that contend with each other. It can't be contained by other cliques.

Definition 3 max-min fairness of time slot: There are N flows in a network, if an end-to-end flow rate allocation vector $(r_1, r_2, \cdots, r_N)$ is feasible and is said max-min fair only if for any flow f_i, its time share cannot be increased without decreasing the fraction of time for any other flow in the clique.

Proposition 1 Scheduling with algorithm MFF-SCC in networks, if each flow passes through one or more than one cliques, then rate allocation in this network is max-min fair of time slot.

Proof: Consider a network with N flows. Each flow passes through one or more than one clique. Let r_k denote the allocated rate of flow k $(k \in \{1, 2, . . ., N\})$. Then there is a feasible rate allocation vector $(r_1, r_2, \cdots, r_N)$.

When flow f_i increases its time share r_i in clique A, but doesn't decrease that of other flow f_k.

On the assumption that $r_i, \cdots, r_j$ share link l in clique A, we consider two different situations:

First to r_k, where $k \in (i, \cdots, j)$ but $k \neq$ i, r_i increases its value to r_i' in clique A, the value of r_k doesn't decrease, then we have

$$r_i' + \sum_{k\in(i+1,\dots j)} r_k > \sum_{k\in(i,\dots j)} r_k \qquad (11)$$

But link l's time share in clique A doesn't increase. According to scheduling algorithm introduced above, it is impossible that r_k increases its share but link l doesn't.

Then to r_k, but k does not belong to $(i, \cdots, j)$, when f_i increases its share in clique A, but f_k doesn't decrease. We would have

$$r_l' + \sum_{k\in(i+1,\dots j)} r_k > \sum_{k\in(i,\dots j)} r_k > C \qquad (12)$$

Where C denote the transmission capability of clique A. That is impossible either.

Following definition 3, allocation vector $(r_1, r_2, \cdots, r_N)$ is max-min fair of time slot to all flows. □

IV. SIMULATION RESULTS

Evaluation is done with ns-2. We develop our MFF-SCC algorithm based on 802.11Ext module. We considered networks as shown in Figure 2 and Figure 5. The basic access method is set in DCF since the similar result with RTS/CTS is show in our experiments. Routing protocol AODV is used all along the simulation. The value of parameter *Max_Q* is set to 5.

We measure the good put achieved by each flow in the given topology by MFF-SCC, and compare the good put with Back Pressure algorithm. Instead of realizing the complex signaling protocol of Back Pressure algorithm, we compute the backlog of queues on neighboring nodes directly. This method decreases the complexity of the algorithm, so we believe it will bring positive effect to its performance.

A. Experiment 1

We conduct this experiment on the network topology as show in Figure 2. Neighboring eight nodes form a network in a line. Flow 1 traverses all eight nodes, and flow 2、3 the other. All sources run FTP which is TCP below and can adjust its rate according to congestion notification. The experiment runs for 150 seconds. We choose data acquired between 11-110 second and plot as show in Figure 3 and 4. As show in Figure 3, the performance of MFF-SCC is good. It assigns max-min fair rates to all the flows, and is within 9.6% of the optimal. MFF-SCC assigns comparable rates to flow 2 and flow 3 as they traverse the same number of nodes. Flow 1 traverses more nodes than the other two flows, so it has longer delay. It acquires less bandwidth than the others. Figure 4 shows the results under Back Pressure algorithm. It starves the flow 1 which traversing the longest path. Flow 1 acquires bandwidth only about 41Kb. There is obvious bias to flows with longer delay.

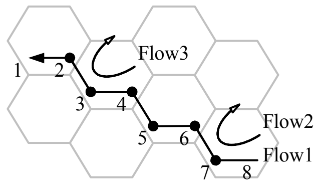

Figure 2. Topology for experiment 1

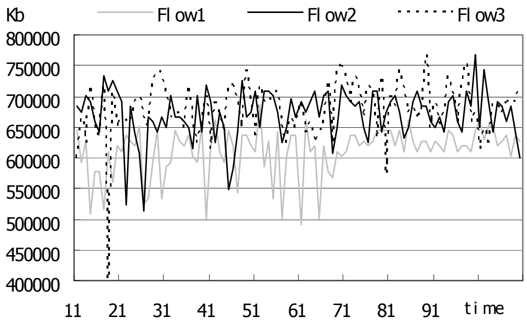

Figure 3. Sharing under MFF-SCC

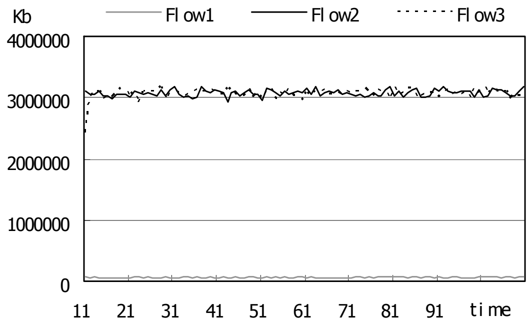

Figure 4. Sharing under Back Pressure

B. Experiment 2

We conduct this experiment on the network topology as show in Figure 5. As shown in Figure5, nodes 4、5、9、10、12 and 13 form a clique of the network, and link (9, 10) is a bottle neck of this sub networks. Flows 1 to 6 traverse this clique. Flow 1、2、3 and 5 run FTP sources. Flow 4 and 6 run CBR (Const Bit Rate) sources. Flow 1、2、3 and 4 leave network on node 13. Flow 5 and 6 leave network on node 6. The experiment runs for 150 seconds under both algorithms. We plot the result as shown in Figure 6. MFF-SCC assigns max-min fair rates to all the flows, and is within 8% of the optimal. But Back Pressure algorithm assigns rates with deviation up to 50%. The throughputs of both algorithms are close, one for 787Kbs and the other 828Kbs, deviation is 2%.

Figure 7 shows the delay of flows in the bottleneck. Larger *Max_Q* values increase the delay. When *Max_Q* set to 7、5 and 3, the average delay are 0.1201、0.1122 and 0.1035 second respectively. Different *Max_Q* values don't interfere with the fairness of MFF-SCC algorithm. The allocation bias keeps under 10%.

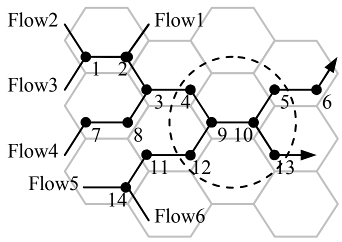

Figure 5. Topology for experiment 2

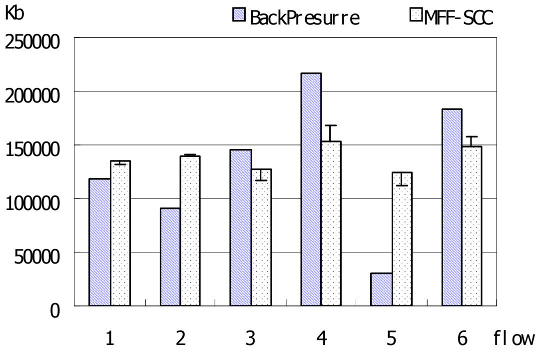

Figure 6. Sharing comparision

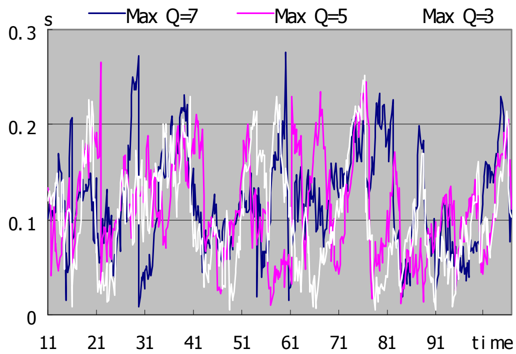

Figure 7. Delays with diferent Max_Q values

V. RELATED WORKS

Efficient scheduling in multi-hop wireless networks has been the subject of intensive research in the past few years [2,4,5]. Early works include those done by Tassiulas and Ephremides[12]. They developed a programmatic procedure for establishing the stability of queuing networks and scheduling policies. And for a Markovian system, they show their method establishes positive recurrence and the existence of a steady-state probability distribution.

To proportion fairness problem in wireless networks scheduling, there has also been a lot of prior work that considers it [8,9]. Proportional fair bandwidth allocations for elastic traffic among a set of mobile users was first presented in [8]. The dynamic congestion control algorithm, which called Greedy Primal-Dual (GPD) algorithm, proves its asymptotic optimality. Umut Akyol [6] studied the signaling problem of jointly performing scheduling and congestion control so that network queues remain bounded and the resulting flow rates satisfy an associated network utility maximization problem. They proposed an algorithm named wireless GPD can be implemented in practice with minimal disruption to existing wireless protocols. GPD algorithm and its following researches improve the fairness of scheduling. But they require exchanging queue length of neighbor nodes to achieve centralized information. Then its computational complexity is too prohibitive for practice [1].

People also do researches based on maximal scheduling. Recently, LQF scheduling has been shown to yield a much larger stability region than the worst case. An LQF scheduler produces a maximal independent set greedily following a sequence according to the queue length order of each link during maximal scheduling. It is shown that in certain circumstance, where the topology satisfies the local-pooling" condition, LQF is optimal. QiaoLi [2] improves the performance of LQF to support QoS in wireless multi-hop networks. They generalize LBS to prioritized maximal scheduling, where in each time slot the scheduler chooses a maximal schedule following a sequence specified by a priority vector, which may be the same as the queue length orders. Therefore, it can achieve the fairness among different links, if the priorities are properly chosen. But they didn't consider the fairness between flows sharing the same output link.

Other than proportion fairness, researchers also investigate max-min fair allocation of bandwidth in wireless networks [10,11]. In [10] formalized the max-min fair objective under wireless scheduling constraints, and present a necessary and sufficient condition for max-min fairness of a bandwidth allocation. And they proposed an algorithm that assigns weights to the transportation sessions dynamically. Each node allocates service tokens to the sessions traversing the node in a round-robin-like fashion. In any time slot, a session would receive a service token does not exceed that received in an end nodes. Nodes do scheduling according session token. Algorithm proposed by Leandros sound reasonable. But its implementation is difficult due to signaling complexity. Prasanna[5] proposed an similar algorithm. It combined maximal scheduling and Token based flow control, and characterized the throughput region attained by maximal scheduling in terms of the interference degree of the network.

As mentioned before, most of researches don't consider the dynamics during scheduling. Some of them simply allocate bandwidth based on node transmission capability or fluctuation of queue length. Others require end users to adapt their rates with certain utility function. Without end user's cooperation the network can't maximize its throughput. Most of them must keep separate queues for different traffic flows and need centralized decision, though with poor scalability.

VI. CONCLUSIONS

We propose a Max Flows First-Shadow queue based Congestion Control (MFF-SCC) algorithm for combined scheduling and congestion control that aims to solve max-min fairness problem. Algorithm does scheduling according flow numbers of each link. And does congestion control based on shadow queue length of each flow. The shadow queue system allows each node to maintain a single FIFO queues for each of its outgoing links, instead of keeping a separate queue for each flow in the network. So flow based scheduling improves fairness of sharing, and shadow queue management improves the performance of the algorithm.

We discuss fairness of the proposed algorithm. Via ns2 simulations show that MFF-SCC significantly outperforms particular allocation algorithm like the back-pressure algorithm. One direction of future works is detailed analysis of the tradeoff between fairness and throughput of the scheduling algorithm. Another is the combination of MFF-SCC and routing. They are all interesting topics for following researches.

REFERENCES

[1] Loc Bui, R.Srikant and Alexander Stolyar. "Novel Architectures and Algorithms for Delay Reduction in Back-pressure Scheduling and Routing," in Proceedings of IEEE INFOCOM2009, Brazil, April 2009.

[2] Qiao Li and Rohit Negi. "Scheduling in Multi-hop Wireless Networks with Priorities," in Proceedings of IEEE INFOCOM2009, Brazil, April 2009.

[3] Wang, P.Jiang, H.Zhuang and W.Poor, H.V. "Redefinition of max-min fairness in multi-hop wireless networks," IEEE Trans. Wireless Commun, vol.7.no.12, pp. 4786-4791, December 2008.

[4] H. T. Cheng and W. Zhuang, "An optimization framework for balancing throughput and fairness in wireless networks with QoS support," IEEE Trans. Wireless Commun., vol. 7, no. 2, pp. 584-593, Feb. 2008.

[5] P.Chaporkar, K. Kar, L.Xiang, and S.Sarkar, "Throughput and Fairness Guarantees Through Maximal Scheduling in Wireless Networks", IEEE Transactions on Information Theory, vol. 54, no. 2, pp572-594, Feb. 2008.

[6] U. Akyol, M. Andrews, P. Gupta, J. Hobby, I. Saniee, and A. L. Stolyar, "Joint scheduling and congestion control in mobile ad-hoc networks," in Proceedings of IEEE INFOCOM, Phoenix, AZ, April 2008.

[7] Stolyar, and Alexander, "Greedy primal-dual algorithm for dynamic resource allocation in complex networks," Queueing Systems, vol. 54, no. 3, pp. 203–220, 2006.

[8] A. Stolyar, "Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm," Queueing Systems, vol. 50, no. 4, pp. 401–457, August 2005.

[9] A. Eryilmaz and R. Srikant, "Fair resource allocation in wireless networks using queue-length based scheduling and congestion control," in Proceedings of IEEE INFOCOM, vol. 3, Miami, FL, March 2005,

[10] L. Tassiulas and S. Sarkar, "Maxmin fair scheduling in wireless adhoc networks,"IEEE Journal on Selected Areas in Communications, vol. 23, No. 1,pp. 163–173, January 2005.

[11] X. L. Huang and B. Bensaou, "On max-min fairness and scheduling in wireless ad-hoc networks: analytical framework and implementation," in Proc. ACM MOBIHOC'01, pp. 221-231, Long Beach, CA, Oct. 2001.

[12] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," IEEE Transactions on Automatic Control, vol. 37, pp. 1936–1948, December 1992.