Past Queue Length Based Low-Overhead Link Scheduling in Multi-beam Wireless Mesh Networks

Arpan Chattopadhyay and A. Chockalingam
Department of ECE, Indian Institute of Science, Bangalore 560012, INDIA

***Abstract*— Wireless mesh networks with multi-beam capability at each node through the use of multi-antenna beamforming are becoming practical and attracting increased research attention. Increased capacity due to spatial reuse and increased transmission range are potential benefits in using multiple directional beams in each node. In this paper, we are interested in low-complexity scheduling algorithms in such *multi-beam* wireless networks. In particular, we present a scheduling algorithm based on queue length information of the past slots in multi-beam networks, and prove its stability. We present a distributed implementation of this proposed algorithm. Numerical results show that significant improvement in delay performance is achieved using the proposed multi-beam scheduling compared to omni-beam scheduling. In addition, the proposed algorithm is shown to achieve a significant reduction in the signaling overhead compared to a current slot queue length approach.**

I. Introduction

Wireless networks are increasingly being considered in several network scenarios as good alternatives to their wireline counterparts, mainly because they avoid cumbersome wiring/-cabling and support user mobility. Recently, due to advancements in multi-antenna beamforming technology, wireless nodes with multiple antennas with beamforming capabilities are becoming practical [1]. Beamforming techniques offer the advantages of increased range compared to omni-directional antenna transmission, and interference avoidance due to the use of narrow beams. The ability to provide several directional beams simultaneously in a given wireless node can offer significant increase in network throughput through spatial reuse. Given the multi-beam capability in each node, it is of interest to address higher layer issues including link scheduling, which forms the focus in this paper.

Capacity of wireless networks with nodes having omni-directional antennas has been studied by Gupta and Kumar in [2]. The capacity of wireless networks can significantly improve if directional antennas are employed at each node, due to spatial reuse and increased transmission range. The capacity of wireless mesh networks with directional antennas has been analyzed in [3]-[5]. Several transmission protocols for wireless mesh networks using directional antennas have been proposed in the literature, e.g., [6]-[8]. Analytical studies reported in the multi-beam networks area are limited. In our work in this paper, we consider a low-complexity queue-length based scheduling strategy suited for wireless networks with multiple beams in each node, and analyze its stability. The proposed scheduling scheme is based on queue length information of past slots.

This work in part was supported by Indo-French Centre for the Promotion of advanced Research (IFCPAR) Project No. 4000-1T-1.

A. Related Work

In [9], Tassiulas and Ephremides derived a centralized joint routing and scheduling policy, known as maximum weight policy, and characterized its stability region. But this policy requires a centralized implementation and global queue length information at the scheduler. So it is prohibitive for practical implementation. Subsequently, several distributed policies have been proposed in the literature. In [10], Tassiulas proposed a policy for switch scheduling to reduce the complexity while stabilizing all arrival rates in the capacity region. A distributed implementation of [10] for wireless networks was proposed by Modiano *et al* in [11], with the weight comparison between matchings being done via an averaging mechanism. Another policy for implementing [10] was studied in [12]. Lin and Shroff, in [13], studied the effect of imperfect scheduling in wireless networks. An issue with the policies in [10],[11],[12] is that the network overhead grows with the network size. So a question of interest is: can we achieve full throughput with low overheads? Lin and Rasool, in [14], showed that close to 1/3 of the capacity region can be achieved with $O(1)$ overhead, i.e., constant overhead. Gupta *et al* in [15], and Joo and Shroff in [16] built on this result in [14] and achieved close to 1/2 the capacity region with constant overhead. In [17], it has been shown that full throughput can be achieved with constant overhead when we consider a node exclusive spectrum sharing model for interference. All these works do not consider multiple beams at the network nodes. Our focus here is on scheduling policies in the context of multi-beam wireless networks.

B. Contribution

Our contribution is this paper can be summarized as follows.

- We present a low-complexity scheduling algorithm for multi-beam networks which takes into account the past slot queue length information at the nodes. The algorithm in [10] was proposed for switch scheduling and deals with current slot queue length information. We extend this algorithm for multibeam network by considering past queue length information.
- We prove the stability of the proposed algorithm through a mean drift analysis.
- We present a distributed implementation of this algorithm. Numerical results show that the proposed algorithm achieves significant delay performance improvement compared to omni-beam scheduling. In addition, the signaling overhead is shown to be significantly reduced in the proposed multi-beam scheduling.

978-1-4244-7138-6/10/$26.00 ©2010 IEEE

II. System Model

Consider a wireless mesh network satisfying the following properties:

1) There are N nodes in the network, each node having B beams. Figure 1 shows an illustrative example of the beam pattern model in a node having $B = 4$ beams.
2) Time is divided into equal-sized slots.
3) Beams are activated/deactivated at the slot boundaries, based on the scheduling policy.
4) Beams are half-duplex; i.e., each beam can be activated either in transmit mode or in receive mode in a slot.
5) All beams in a node can be activated simultaneously, if the schedule permits.
6) There are J flows in the network, each designated by its destination node. A flow can have packet arrivals from every node in the network other than its destination node.
7) In each node in the network, there are J buffers (queues), one for each flow.
8) A *physical link* between two nodes is said to exist if they are within each others decodable range. Let L denote the number of physical links in the network.
9) A physical link between two nodes is said to be activated in a slot if they beamform towards each other using a transmit beam in one node and a receive beam in the other node in that slot.
10) Each physical link is viewed as two *directional links* one in each direction; e.g., a physical link between nodes A and B is viewed as one directional link from A to B and another from B to A, and only any one of them can be activated in a slot.
11) Each directional link is viewed as J *virtual links*, one for each flow. Therefore, for a given physical link, there will be $2J$ virtual links, and only any one of these virtual links can be activated in a slot.
12) The transmission rates on virtual links in the entire network is represented by a $J \times 2L$ matrix. The feasible set of rate matrices, denoted by Γ, is assumed to have a finite number of elements.
13) The $J \times 2L$ *rate matrix* in slot t is denoted by $R(t)$. A simple and widely used model considers $R_{jl} \in \{0, 1\}$. In a more general setting, the R_{jl}'s can be based on the link SINRs.
14) The feasible set, Γ, can differ in omni-beam and multi-beam scenarios due to spatial reuse. For e.g., in a 3-node linear network with one flow, assuming node exclusive spectrum sharing for omni-beam case [17] and assuming non-directional links with $R_{jl} \in \{0, 1\}$, the feasible set of rates on links 1 and 2 for the case of omni-beam is $\Gamma = \{(0, 0), (0, 1), (1, 0)\}$. For the case of 2 beams per node, $\Gamma = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$.
15) Packets sent on links that are activated as per a feasible rate matrix are assumed to be decoded error-free.
16) Let $Q(t)$ denote the $J \times N$ queue length matrix in slot t, such that $Q_{j,n}(t)$ denotes the queue length at nth node corresponding to jth flow in the tth slot.
17) Let P denote the $2L \times N$ adjacency matrix of the network, where

$$P_{l,n} = \begin{cases} 1 & \text{if directional link } l \text{ originates from node } n \\ -1 & \text{if directional link } l \text{ is incident on node } n \\ 0 & \text{if directional link } l \text{ is not connected with node } n. \end{cases}$$

18) Finally, let $A_{j,n}(t)$ denote the number of new packet arrivals at the nth node corresponding to the jth flow in the tth slot.

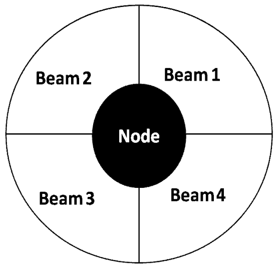

Fig. 1. An illustrative example of the beam pattern model in a 4-beam node.

A. Algorithm Based on Current Slot Queue Length (CSQL)

Optimum scheduling policy based on maximum weight matching achieves full capacity in wireless networks [9], but it is less practical because of its high computational complexity. Several sub-optimal scheduling algorithms for wireless networks have been proposed in the literature; e.g., the low-complexity algorithm proposed in [10] is based on $Q(t)$, the global queue length information of the current slot, whose worst case overhead is $O(N^2)$. We have adapted this current slot queue length (CSQL) information based algorithm for the multi-beam system model described above as follows.

Let us assume that at slot t, the scheduler knows $Q(t)$ and $R(t-1)$. We need to decide the schedule for slot t, i.e., determine $R(t)$. Let $R^\star(t)$ be the feasible rate matrix which maximizes global weighted sum of link rates when queue length matrix is $Q(t)$, i.e.,

$$R^\star(t) \;=\; \arg\max_{R} \; tr\{RPQ^T(t)\}, \qquad (1)$$

where $tr\{.\}$ denotes the trace operator, and $PQ^T(t)$ is the transpose of the $J \times 2L$ *weight matrix*, where the weight of a virtual link for a flow is defined as the difference between the corresponding source and destination queue lengths.

1) In slot t, pick some feasible rate matrix R randomly from Γ such that $\Pr\{R = R^\star(t)\} \geq \delta, \quad 0 < \delta < 1$.
2) Choose $R(t) = R$ if $tr\{RPQ^T(t)\} \geq tr\{R(t-1)PQ^T(t)\}$, i.e., R offers higher weighted sum of link rates compared to $R(t-1)$ for $Q(t)$; else, $R(t) = R(t-1)$.

Motivated to reduce the high signaling overhead involved in the implementation of this CSQL based scheduling, we consider the following algorithm which computes schedules based on the queue length information of the past slots.

B. Algorithm Based on Past Slots Queue Length (PSQL)

Let us now assume that, instead of knowing the CSQL information $Q(t)$, the scheduler knows the queue length information and rate matrices of the past D slots; i.e., the scheduler knows $Q(t-D)$ and $R(t-D)$, where D is a finite positive integer. In this past slots queue length (PSQL) based scheme, the schedule for slot t, $R(t)$, is obtained as follows.

1) In slot t, pick some feasible rate matrix R randomly from Γ such that $\Pr\{R = R^\star(t)\} \geq \delta, \quad 0 < \delta < 1$, and $R^\star(t) = \arg\max_R \; tr\{RPQ^T(t-D)\}$.
2) Choose $R(t) = R$ if $tr\{RPQ^T(t-D)\} \geq tr\{R(t-D)PQ^T(t-D)\}$, i.e., R offers higher weighted sum of link rates compared to $R(t-D)$ for $Q(t-D)$; else, $R(t) = R(t-D)$.

In the following section, we prove the stability of this PSQL algorithm through a mean drift analysis.

III. Stability of the PSQL Algorithm

Consider the random process

$$X_n = \{R(nD-1), .., R(nD-D), Q(nD-D), .., Q(nD-2D)\} \quad (2)$$

where $n = 0, 1, 2, \cdots$. It can be observed that X_n is an irreducible discrete-time Markov chain (DTMC) The PSQL algorithm will achieve full capacity, i.e., it will stabilize all queues in the network for all set of stabilizable arrival rates if the DTMC X_n is positive recurrent. The queue lengths in the network evolve as

$$Q(t+1) \;=\; Q(t) + A(t) - R(t)P. \quad (3)$$

To carry out the drift analysis, we consider the following Lyapunov function

$$h(X_n) \;=\; f(X_n) + \frac{2}{\delta^D} g(X_n), \quad (4)$$

where

$$f(X_n) \;=\; tr\{Q^T(nD-D)Q(nD-D)\} \quad (5)$$

and

$$g(X_n) \;=\; tr\Big\{\Big[R^\star(nD-1) - R(nD-1)\Big]PQ^T(nD-D-1) + \cdots + \Big[R^\star(nD-D) - R(nD-D)\Big]PQ^T(nD-2D)\Big\}. \quad (6)$$

$f(X_n)$ is a non-negative function on the state space. Now,

$$\begin{aligned}\mathbb{E}\big[f(X_{n+1}) - f(X_n)\big|X_n\big] \;&=\; \mathbb{E}\,tr\Big\{\big[Q(nD) - Q(nD-D)\big] \big[Q(nD) + Q(nD-D)\big]^T\Big|X_n\Big\} \\ &\leq\; E\,tr\Big\{\Big[A(nD-D) - R(nD-D)P + \cdots + A(nD-1) - R(nD-1)P\Big]\Big[2Q(nD-D) + A(nD-D) - R(nD-D)P \\ &\quad + \cdots + A(nD-1) - R(nD-1)P\Big]\Big|X_n\Big\}. \end{aligned} \quad (7)$$

The inequality in (7) is due to the queue length for a flow at its destination being zero. Now, the total number of arrivals and departures over D slots can be bounded above. Hence,

$$\mathbb{E}\big[f(X_{n+1}) - f(X_n)\big|X_n\big] \leq \text{ const} + 2\mathbb{E}\,tr\Big\{\Big[A(nD-D) - R(nD-D)P + \cdots + A(nD-1) - R(nD-1)P\Big]Q^T(nD-D)\Big|X_n\Big\}.$$

For stabilizable mean arrival rate, $\mathbb{E}[A(nD-D)] \leq \sum_k c_k R'_k P$, where the entries of R'_k are same as those in $R_k \in \Gamma$, except at entries which correspond to virtual links incident on the corresponding destination nodes, where they are made as zeros. Also, $0 < c_k$ and $\sum_k c_k < 1$. Hence,

$$\begin{aligned}&\mathbb{E}\,tr\Big\{\Big[A(nD-D) - R(nD-D)P\Big]Q^T(nD-D)\Big|X_n\Big\} \\ &\leq \text{ const} + \mathbb{E}\,tr\Big\{\Big(\sum_k c_k - 1\Big)R^\star(nD-D)PQ^T(nD-2D)\Big|X_n\Big\} \\ &+ \mathbb{E}\,tr\Big\{\Big[R^\star(nD-D) - R(nD-D)\Big]PQ^T(nD-2D)\Big|X_n\Big\}, \end{aligned} \quad (8)$$

where we have used the fact that $R^\star(nD-D)$ provides the optimum weight for $Q(nD-2D)$. Following similar steps for other terms, we obtain

$$\mathbb{E}\big[f(X_{n+1}) - f(X_n)|X_n\big] \leq \text{ const} + 2g(X_n) - \epsilon\, tr\Big\{R^\star(nD-D) PQ^T(nD-2D) + \cdots + R^\star(nD-1)PQ^T(nD-D-1)\Big\}, \quad (9)$$

where ϵ is strictly positive. Now, letting $\mathcal{A}$ to denote the event $\big[R(nD), .., R(nD+D-1)\big] \neq \big[R^\star(nD), .., R^\star(nD+D-1)\big]$, we can write

$$\begin{aligned}\mathbb{E}\left\{g(X_{n+1})|X_n\right\} &= \Pr\left\{\mathcal{A}\right\}\mathbb{E}\,tr\Big\{\Big[R^\star(nD) - R(nD)\Big]PQ^T(nD-D) \\ &+ \cdots + \Big[R^\star(nD+D-1) - R(nD+D-1)\Big]PQ^T(nD-1)\Big|X_n, \mathcal{A}\Big\} \\ &\leq (1-\delta^D)\,\mathbb{E}\,tr\Big\{\Big[R^\star(nD) - R(nD)\Big]PQ^T(nD-D) + \cdots \\ &+ \Big[R^\star(nD+D-1) - R(nD+D-1)\Big]PQ^T(nD-1)\Big|X_n, \mathcal{A}\Big\}. \end{aligned} \quad (10)$$

Now, assuming that the maximum number of arrivals to be bounded and observing that $tr\{R(nD)PQ^T(nD-D)\} \geq tr\{R(nD-D)PQ^T(nD-D)\}$, we obtain

$$\begin{aligned}&\mathbb{E}\,tr\Big\{\Big[R^\star(nD) - R(nD)\Big]PQ^T(nD-D)\Big|X_n, \mathcal{A}\Big\} \;\leq\; \text{const} + \\ &\mathbb{E}\,tr\Big\{\Big[R^\star(nD-D) - R(nD-D)\Big]PQ^T(nD-2D)\Big\}. \end{aligned} \quad (11)$$

Performing similar operations on other terms, we obtain from (10)

$$\mathbb{E}\{g(X_{n+1}) - g(X_n)|X_n\} \;\leq\; \text{const} - \delta^D g(X_n). \quad (12)$$

From (9) and (12), we obtain the mean drift of $h(X_n) = f(X_n) + \frac{2}{\delta^D}g(X_n)$ as

$$\begin{aligned}\mathbb{E}\{h(X_{n+1}) - h(X_n)|X_n\} \;&\leq\; \text{const} - \epsilon\, tr\Big\{R^\star(nD-D) \\ &PQ^T(nD-2D) + .. + R^\star(nD-1)PQ^T(nD-D-1)\Big\}. \end{aligned} \quad (13)$$

We observe that if one among $Q(nD-D-1), \ldots, Q(nD-2D)$ has at least one element sufficiently large, then the R.H.S of (13) will be negative. So the mean drift of $h(X_n)$ is negative over the entire state space of X_n except for a finite number of states where the drift is bounded above. Hence, X_n is a positive recurrent DTMC. Hence this PSQL policy achieves full capacity. We note that the analysis is same for omni-beam and multi-beam case. The only difference is that the set of feasible rate matrices are different in the two scenarios.

IV. Distributed Implementation of the PSQL Algorithm

In this section, we present a distributed implementation of the PSQL algorithm given in Sec. II-B, and show that the pipelined nature of signaling messages over multiple past slots in PSQL significantly reduces the signaling overhead compared to CSQL.

A. Distributed Implementation

At the beginning, the following initialization is done.

1) A maximal spanning tree of the network is found, and each node is provided with the information of the tree structure and the root node.
2) Nodes are colored in such a way that any pair of nodes having the same color can transmit without collision in the same slot. Let M denote the total number of colors in the network.
3) Each node is made aware of its own color, priority of that color, and the number of colors in the network. Since the knowledge of the tree is available at all nodes, each node knows its parent node, child nodes, number of hops from it to its farthest child node as well as to the root node, and the height of the tree, L_{max}.
4) The value of D is also known at all nodes.

Slot Structure: Each slot consists of three parts, namely, control part 1 (CP1), control part 2 (CP2), and data part (DP), as shown below.

CP1	CP2	DP

CP1 is meant for the purpose of directional RTS-CTS exchange between nodes for picking an R matrix (i.e., accomplish Step 1 in the PSQL algorithm), which is implemented in a distributed way. CP1 is split into M pairs of RTS-CTS minislots as shown below, where M is the number of colors in the network.

CP1:

RTS 1	CTS 1	...	RTS M	CTS M

CP2 is meant to accomplish Step 2 of the algorithm in a distributed way using omni-directional transmission of signaling messages in a pipelined manner. CP2 consists of multiple minislots to carry the cumulative weight information towards the root node, and the decision made by the root node towards the leaf nodes. For e.g., if $D = 2L_{max}$, the number of minislots in CP2 can be M, where M is the number of colors in the network. This is because, since the height of the tree is L_{max}, signaling messages should traverse L_{max} hops from the farthest child to the root and vice versa for checking R and broadcasting the decision down the tree. In this case, in each CP2, one minislot for one color is adequate.

DP is meant for directional transmission of data. As per the protocol, the schedule for activating links for DP in slot t is decided by the root node ahead of slot t. For e.g., if $D = 2L_{max}$, then the schedule for slot t is decided at slot $t - \frac{D}{2}$.

Protocol Operation: In the following, we describe the protocol for $D = 2L_{max}$. In RTS 1 minislot, the highest priority nodes (identified by their colors), for each of their beams, either choose no link with some non-zero probability or choose randomly one link from the set of all virtual links outgoing from that beam with a uniform distribution. An RTS message contains the IDs of the sending and the receiving nodes, index of the flow corresponding to the chosen virtual link, current slot queue length of the chosen flow at the sending node[1], and the queue length at the sending node corresponding to the flow that is going to be activated in the DP of the current slot. On receiving the RTS 1 messages, the receiving nodes send their ACKs in the CTS 1 minislot. Similar RTS and CTS message exchanges happen for the remaining $M-1$ colors in the next $M-1$ RTS-CTS minislot pairs. We note that in CTS 2 to CTS M, the messages can be NACK or ACK; NACK is for disallowing granting of any link that can interfere with the transmission of an already assigned link in the previous RTS-CTS minislots in the slot. So, at the end of CP1 of slot $(t-D)$, node n becomes aware of $tr\{RPQ^T(t-D)\}_n$ and $tr\{R(t-D)PQ^T(t-D)\}_n$, i.e., the weighted sum of link rates of all the virtual links incident on node n corresponding to the schedules R and $R(t-D)$. The nth node metric at slot $t-D$, defined as $F_n(t-D) \triangleq tr\{RPQ^T(t-D)\}_n - tr\{R(t-D)PQ^T(t-D)\}_n$ are passed to the root node in a cumulative manner in CP2 over several slots to determine $R(t)$. The root node gets all the information it needs to decide $R(t)$ by the end of slot $t - \frac{D}{2}$. If $\sum_n F_n(t-D) \geq 0$, then the root node decides to switch the schedule to R in slot t, and not to switch otherwise. This decision is conveyed to all the nodes on the downstream in CP2 minislots so that every node comes to know of the schedule to use at slot t.

[1] We note that this CSQL information will become PSQL information when it reaches the root after several slots for performing Step 2 of the algorithm.

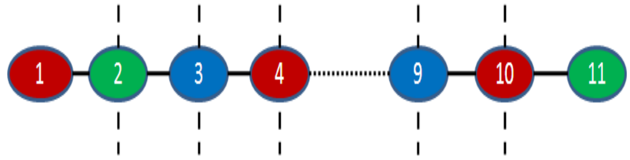

Fig. 2. 11-node linear network with 2 beams per node.

An Illustration: Consider the 11-node linear network with 3 colors (i.e., $M = 3$) shown in Fig. 2, with node 1 as the root node so that $L_{max} = 10$. The colors arranged in descending order of priority are Green, Red, Blue. Consider $D = 2L_{max} = 20$. In the 1st minislot of CP2 of slot $(t-19)$, node 11 will send $F_{11}(t-20)$ to node 10. Node 10 will compute $F_{11}(t-20) + F_{10}(t-20)$ and send it to node 9 in minislot 2 of CP2 of slot $(t-18)$. Node 9 will compute $F_{11}(t-20) + F_{10}(t-20) + F_9(t-20)$ and send it to node 8 in minislot 3 of CP2 in slot $(t-17)$. This cumulative transfer of node metric continues so that in slot $(t-10)$, node 2 will send $\sum_{n=2}^{11} F_n(t-20)$ to the root node 1. The root node computes $\sum_{n=1}^{11} F_n(t-20)$ and decides on the schedule to be used in slot t. This decision is relayed to all nodes in slots $(t-9)$ to t, which will be used by all nodes in slot t .

Signaling Overhead: We find that the signaling overhead size per slot is $O(M)$ for this distributed implementation. For large networks with efficient coloring, M can be much smaller than N. So, the proposed protocol achieves full capacity with low signaling overhead. In a fully connected network, we encounter the worst case overhead, where $M = N$ and the overhead becomes $O(N)$.

Comparison between PSQL and CSQL in Multi- vs Omni-beams: In the 11-node linear network in Fig. 2, the overhead is $3 + (3 \times 20) = 63$ messages for CSQL, because the schedule for slot t must be found in slot t itself. So CP2 in CSQL requires 20 hop overhead exchange, which is multiplied by

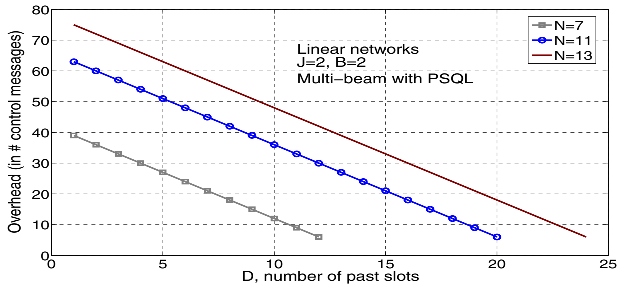

Fig. 3. Signaling overhead as a function of D in the proposed distributed implementation of the PSQL algorithm. Linear networks with $N = 7, 11, 13$. $J = 2$, $B = 2$.

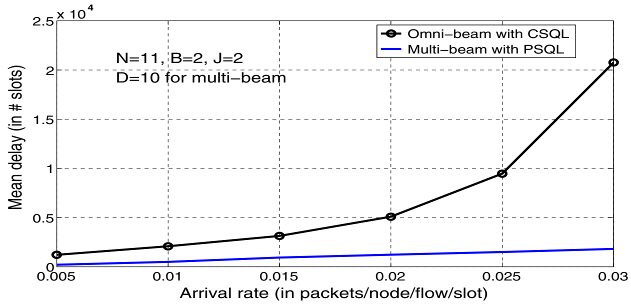

Fig. 4. Mean delay performance comparison between PSQL multi-beam and CSQL omni-beam in 11-node linear network. $B = 2$ and $D = 10$ for PSQL multi-beam. $J = 2$.

3 due to 3 colors. For a general network, the overhead required in CSQL is $O(MN)$. On the other hand, the PSQL the overhead reduces from 63 messages to 6 messages when we use $D = 20$ for the linear network Fig. 2. Figure 3 illustrates the reduction in the overhead as a function of D; larger the value D, larger is the overhead reduction. This reduction comes along with an increased degradation in the delay performance for increasing D (because of ignoring current slot queue length information), which is shown in the simulated mean delay performance plots in Figs. 4 and 5. However, it can be seen that with multi-beam capability ($B = 2$), the proposed PSQL algorithm outperforms significantly in delay performance the omni-beam scheme employing even CSQL . We also compared the delay performance between CSQL and PSQL algorithms, both under multi-beam case, for a 16-node grid network. The results are shown in Fig. 6. While CSQL provides better delay performance compared to PSQL, the overhead of PSQL is significantly less compared to CSQL.

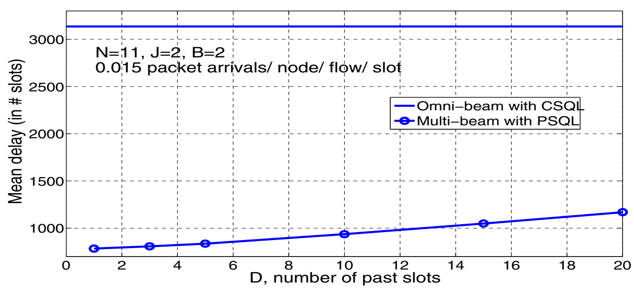

Fig. 5. Mean delay performance comparison between PSQL multi-beam for different D and CSQL omni-beam in 11-node linear network. $B = 2$ for PSQL multi-beam. $J = 2$, arrival rate = 0.015 packets/node/flow/slot.

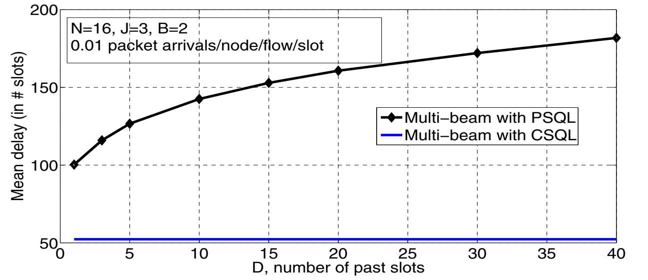

Fig. 6. Mean delay performance comparison between PSQL multi-beam for different D and CSQL multi-beam in 16-node grid network. $B = 2$ for PSQL multi-beam. $J = 3$, arrival rate = 0.01 packets/node/flow/slot.

V. Conclusions

We presented a low-complexity scheduling algorithm for multi-beam wireless mesh networks. The algorithm obtained the schedules based on the queue length information of the past slots. The proposed algorithm was proved to achieve full capacity. Delay performance in multi-beam case with the proposed scheduling was shown to be significantly better that in omni-beam case based on current slot queue length information. The proposed scheme also resulted in significant reduction in signaling overhead compared to current slot queue length based scheduling.

References

[1] J. M. Gilbert, C. H. Doan, S. Emami, and C. B. Shung, "A 4-Gbps uncompressed wireless HD A/V transceiver chipset," *IEEE Micro*, pp. 56-64, 2008.

[2] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. on Information Theory*, vol. 46, no. 2, pp. 388-404, 2000.

[3] S. Yi, Y. Pei, and S. Kalyanaraman, "On the capacity improvement of ad-hoc wireless networks using directional antennas," *MobiHoc'2003*.

[4] J. Zhang and X. Jia, "Capacity analysis of wireless mesh networks with omni or directional antennas," *IEEE INFOCOM'2009*, April 2009.

[5] H-N. Dai, K-W. Ng, R. C-W. Wong, and M-Y. Wu, "On the capacity of multi-channel wireless networks using directional antennas," *IEEE INFOCOM'2008*, April 2008.

[6] K. Sundaresan, K. Ramachandran, and S. Rangarajan, "Optimal beam scheduling for multicasting in wireless networks," *MobiCom'2009*.

[7] H. Jasani and K. Yen, "Performance improvement using directional antennas in ad-hoc networks," *IJCSNS Intl. Jl. of Computer Science and Network Security,* vol. 6, no. 6, pp. 180-188, June 2006.

[8] F. Dai and J. Wu, "Efficient broadcasting in ad-hoc wireless networks using directional antennas," *IEEE Trans. on Parallel and Distributed Systems*, vol. 17, iss. 4, pp. 335-347, April 2006.

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

[10] L. Tassiulas, "Linear complexity algorithms for maximum throughput in radio networks and input queued switches," *Proc. IEEE INFOCOM'1998*, March-April 1998.

[11] E. Modiano, D. Shah, and G. Zussman, "Maximizing throughput in wireless networks via gossiping," *Proc. ACM SIGMETRICS/IFIP Performance*, vol. 34, iss. 1, pp. 27-38, June 2006.

[12] A. Eryilmaz, A. Ozdaglar, and E. Modiano, "Polynomial complexity algorithms for full utilization of multi-hop wireless networks," *IEEE INFOCOM'2007*, May 2007.

[13] X. Lin, N. B. Shroff, "The impact of imperfect scheduling on cross-layer rate control in wireless networks," *Proc. IEEE INFOCOM'2005*, March 2005.

[14] X. Lin and S. Rasool, "Constant-time distributed scheduling policies for ad-hoc wireless networks," *Proc. IEEE CDC'2006,* Dec. 2006.

[15] A. Gupta, X. Lin, and R. Srikant, "Low-complexity distributed scheduling algorithms for wireless networks," *Proc. IEEE INFOCOM'2007*, May 2007.

[16] C. Joo and N. Shroff, "Performance of random access scheduling schemes in multi-hop wireless networks," *ACSSC '06*, Oct-Nov 2006 .

[17] S. Sanghavi, L. Bui, and R. Srikant, "Distributed link scheduling with constant overhead," *ACM SIGMETRICS'07*, June 2007.