Node-based Scheduling with Provable Evacuation Time

Bo Ji and Jie Wu
Department of Computer and Information Sciences
Temple University, Philadelphia, PA
Email:{boji, jiewu}@temple.edu

***Abstract*—This paper studies the link scheduling problem in multihop wireless networks without future packet arrivals, with an aim of minimizing the time interval needed for evacuating all the existing packets. We consider the scenario with single-hop traffic flows under the one-hop interference model. In this setting, the minimum evacuation time problem is equivalent to the classic multigraph edge coloring problem, which is generally NP-hard. Although many approximation algorithms have been studied, almost all of them require computing the schedules (or the colors) all at once, and thus have a complexity that is dependent on the number of packets in the network. This, however, renders these algorithms unsuitable for the considered application of link scheduling for packet evacuation, as they typically incur an impractically high complexity when there are a large number of initial packets waiting to be transmitted, even if the underlying network has a small size (i.e., the node count and the link count). Instead, it is desirable to compute the schedules (or the colors) in an online fashion, i.e., to quickly compute one schedule at a time. Unfortunately, none of the existing online algorithms can guarantee an approximation ratio better (or smaller) than 2. To that end, in this paper we suggest two scheduling algorithms using a node-based approach, and prove that their approximation ratios are both no worse (or greater) than $\frac{3}{2}$. In addition, these scheduling algorithms can also serve as an alternative for achieving Shannon's bound, which is well known in the graph coloring literature.**

***Index Terms*—Link scheduling, multihop wireless networks, evacuation time, node-based approach, provable performance guarantees, multigraph edge coloring.**

I. Introduction

How to design efficient link scheduling algorithms for multihop wireless networks has been a vital and challenging problem (see [1]–[3] and the references therein). In this paper, we focus on the setting with existing packets in the network and without future packet arrivals. The performance metric of interest is the *evacuation time*, defined as the time interval needed for completely draining all the existing packets in the network. Clearly, the evacuation time is a critical performance metric in the setting without arrivals. Moreover, in the setting with arrivals, the evacuation time can also be viewed as a measure of the short-term throughput, which is closely related to the delay performance. In particular, a scheduling algorithm being evacuation-time-optimal in the settings without arrivals is known to be a necessary condition for sample-path delay optimality[1] in input-queued switches [5].

In this paper, we assume that the traffic flows are single-hop, i.e., the packets waiting to be transmitted over a link, will leave the network once they are transmitted. We also assume the *one-hop* (or *node-exclusive*) interference model [2], [6], [7], where two links sharing a common end node cannot be activated to transmit at the same time. In this setting, the minimum evacuation time problem is equivalent to the classic multigraph edge coloring problem. In contrast to a simple graph that allows at most one edge between any two vertices, in a multigraph, two vertices may be connected by more than one edge, which is called a multi-edge. The edge coloring problem is concerned with finding the minimum number of colors, also called chromatic index, so that each multi-edge is assigned a color and no two multi-edges sharing a common end node have the same color. This is a well-known NP-hard problem in general [8]. Hence, many approximation algorithms have been proposed and analyzed in the edge coloring literature (see [9] and the references therein).

Nevertheless, almost all of these approximation algorithms rely on the recoloring techniques, and require computing the schedules (or the colors) all at once, and thus have a complexity that is dependent on the number of packets in the network. This, however, renders these algorithms unsuitable for certain types of applications, such as the link scheduling problem that we consider in this paper, as they typically incur an impractically high complexity when there are a large number of initial packets waiting to be transmitted, even if the underlying network has a small size (i.e., the node count and the link count). Instead, it is desirable to compute the schedules (or the colors) in an online fashion, i.e., to quickly compute one schedule at a time. Such algorithms are called online algorithms throughout this paper. For online algorithms, we require that the complexity of computing a color depend on the network size only, and is independent of the number of packets in the system. Note that although we consider a setting without future packet arrivals in this paper, online algorithms will still be functional in the settings with arrivals, while the pure edge-coloring-based algorithms will become less relevant.

To that end, the evacuation time performance of several

[1] A scheduling algorithm is said to achieve sample-path delay optimality if it minimizes the total queue lengths over the network in every time-slot and for any sample path of traffic arrival patterns [4]. This is a very strong notion of delay optimality, and perhaps is also the hardest to attain.

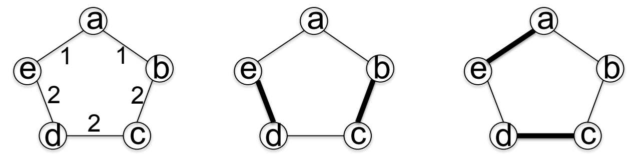
popular online algorithms in the literature of wireless scheduling has been studied [5], [10], [11]. These algorithms either take a link-based approach and make decisions based on the link weight (defined as the queue length at the link), such as Maximum (edge-)Weighted Matching (MWM) [1], MWM-α [12], [13], and Greedy Maximal Matching (GMM) [14], [15], or are queue-length agnostic, such as the Randomized Maximal Matching (RMM) algorithm [14]. In the settings with arrivals, these popular scheduling algorithms are either throughput-optimal, i.e., they stabilize the network under any arrival rates strictly within the network capacity region, or have a provable efficiency ratio, i.e., they achieve a non-trivial and constant fraction of the network capacity region. However, none of them can guarantee an approximation ratio better (or smaller) than 2 for the minimum evacuation time problem.

Very recently, a class of Lazy Heaviest Port First (LHPF) algorithms[2] focusing on scheduling nodes with a heavy weight (or degree, defined as the summed queue length of the links incident to the node), have been studied in [5], [10] (also see [16], [17]) for input-queued switches. Note that an important algorithm called the Maximum Vertex-weighted Matching (MVM) algorithm (described in Section III), which maximizes the sum of the weight of the matched nodes, also belongs to the class of LHPF algorithms. It is remarkable that the LHPF algorithms are both evacuation-time-optimal and throughput-optimal for input-queued switches that can be modeled as bipartite graphs. However, it has largely been an open question *whether one can find online algorithms that guarantee an approximation ratio better (or smaller) than 2 for the minimum evacuation time problem in multihop wireless networks with single-hop traffic flows.*

In this paper, we answer the above question, and make the following contributions.

- First, we prove that the MVM algorithm guarantees an approximation ratio no greater than $\frac{3}{2}$, by showing that under MVM, the maximum node weight (or degree) will decrease by at least two within every three consecutive time-slots.
- Second, we make an important observation: in order to achieve an approximation ratio of $\frac{3}{2}$, it suffices to focus on scheduling the *critical* nodes (i.e., nodes with a maximum weight). Then, by using the insights obtained, we propose a new algorithm that guarantees an approximation ratio no greater than $\frac{3}{2}$ as well, but at a complexity of $O(m\sqrt{n})$, lower than $O(m\sqrt{n}\log n)$ of MVM, where n and m are the node count and the link count, respectively. This is achieved by reassigning bounded integer weights to the nodes, while ensuring that the critical nodes are still given a higher priority.
- Finally, as a byproduct, our proposed algorithms can also serve as an alternative for achieving Shannon's bound [18], which is well known in the graph coloring literature.

[2]Consider any given $\alpha \in (0, 1]$ and any node i that has a weight within α-fraction of the heaviest weight over all the nodes. A matching M is an LHPF matching if node i is not scheduled by M, then there does not exist any other matching M' that can schedule i and all the nodes heavier than i that are scheduled by M. A scheduling algorithm is an LHPF algorithm if it produces an LHPF matching in every time-slot [5].



(a) Topology (b) An MVM (c) A CNM

Fig. 1: A size-5 ring topology. The number of packets at each link is labeled by the link in (a). In this example, the node degrees are $\{2, 3, 4, 4, 3\}$ for nodes $\{a, b, c, d, e\}$; the maximum node degree is 4; nodes c and d are the critical nodes. The node weights of $\{a, b, c, d, e\}$ under MVM are $\{2, 3, 4, 4, 3\}$. Assume $B_1 = 1$ and $B_2 = 2$, then the node weights of $\{a, b, c, d, e\}$ under CNM are $\{1, 1, 2, 2, 1\}$. In (b) and (c), the thick edges denote the links included in the matching. The matching in (b) is an MVM as well as a CNM, while the matching in (c) is a CNM but not an MVM.

II. System Model

We consider a multihop wireless network described by a (simple) graph $G = (V, E)$, where V denotes the set of n nodes/vertices and E denotes the set of m links/edges. Our focus is on a setting where there are initial packets waiting to be transmitted, and there are no future packet arrivals. Time is assumed to be slotted. We assume packets are of unit length, and we let $Q_l(t)$ denote the number of remaining packets that need to be transmitted over link $l \in E$ in time-slot t. Without loss of generality, we assume all the links have at least one packet at the very beginning, i.e., $Q_l(0) > 0$ for all $l \in E$. *In the following time-slots, we remove the links with no remaining packets.* Let $L(i)$ denote the set of links incident to node i. We let $d_i(t) \triangleq \sum_{l \in L(i)} Q_l(t)$ denote the degree of node i in time-slot t accounting for multiplicity, and let $\Delta(t) \triangleq \max_{i \in V} d_i(t)$ denote the maximum node degree in time-slot t. A node having degree $\Delta(t)$ is called a *critical* node. For example, in Fig. 1a, the degrees of nodes $\{a, b, c, d, e\}$ are $\{2, 3, 4, 4, 3\}$; the maximum node degree is 4; nodes c and d are the critical nodes.

We assume that the traffic flows are single-hop, i.e., once a packet is transmitted over a link, the packet leaves the system. In each time-slot, only a subset of links conforming to the *one-hop* (or *node-exclusive*) interference constraint can be activated to transmit packets. Clearly, such a subset of links form a *matching*, which we also call a *schedule*, and is denoted by M. A matching M is called a *maximal matching* if no more links can be added to M without violating the interference constraint. Let $\mathcal{M}$ denote the set of all matchings over graph G. For ease of presentation, we assume unit link capacity, i.e., only one packet can be transmitted over a link if the link is activated. However, our results can also be extended to the general scenario with heterogeneous link capacities by considering the workload defined as $\lceil$number of packets/link capacity$\rceil$.

Let T^S denote the evacuation time of scheduling algorithm S, and let $\mathcal{X}'$ denote the minimum evacuation time over all the algorithms. Note that $\mathcal{X}'$ is also called the *chromatic index* in the graph coloring literature, and depends only on the network topology and initial configuration.

III. MVM Algorithm

In this section, we analyze the evacuation time performance of the MVM algorithm, and prove that MVM guarantees an approximation ratio no greater than $\frac{3}{2}$.

We start by describing the operations of the MVM algorithm. By slightly abusing the notations, we remove the dependence on t, e.g., we use Q_l to denote $Q_l(t)$. In each time-slot, we assign a weight w_i to node i as its degree, i.e., $w_i = d_i$, and let $w(M) \triangleq \sum_{i: M \cap L(i) \neq \emptyset} w_i$ denote the weight of matching M, i.e., the sum of the weight of the nodes matched by M. A matching M^* is an MVM if it maximizes the sum of the weight of the matched nodes, i.e., $M^* \in \mathrm{argmax}_{M \in \mathcal{M}}\, w(M)$. Clearly, an MVM is a maximal matching when all the nodes have positive weights. Fig. 1b shows an example of an MVM in a size-5 ring topology, where the node weights under MVM are equal to the node degrees, which are $\{2,3,4,4,3\}$ for nodes $\{a,b,c,d,e\}$, and the resultant MVM, {(b,c), (d,e)}, has a weight of 14. The MVM algorithm finds an MVM in every time-slot. It is shown in [19] that MVM has a complexity of $O(m\sqrt{n}\log n)$, lower than $O(mn)$ of its edge-weighted counterpart, MWM. Next, we state the main result of this paper in the following theorem.

Theorem 1. *The MVM algorithm has an approximation ratio no greater than $\frac{3}{2}$ for the minimum evacuation time problem.*

Proof. We want to show $T^{\text{MVM}} \leq \frac{3}{2}\mathcal{X}'$. If $\Delta = 1$, this is trivial as $T^{\text{MVM}} = \mathcal{X}' = \Delta$. Now, suppose $\Delta \geq 2$. Then, this follows immediately from 1) Proposition 1: under the MVM algorithm, the maximum node degree decreases by at least two within every three consecutive time-slots, i.e., $T^{\text{MVM}} \leq \frac{3}{2}\Delta$, and 2) an obvious fact: it takes at least Δ time-slots to drain all the packets over the links incident to a node with maximum degree, i.e., $\Delta \leq \mathcal{X}'$. □

Therefore, it remains to prove Proposition 1 stated below.

Proposition 1. *Suppose the maximum node degree is no smaller than two. Under the MVM algorithm, the maximum node degree decreases by at least two within every three consecutive time-slots.*

Before we prove Proposition 1, we restate a useful result of [20] in Lemma 1, and present an important property of MVM in Lemma 2. Both lemmas will be used in the proof of Proposition 1.

Lemma 1 (Theorem 1 of [20]). *Consider a graph G. Suppose that the subgraph of G induced by all the vertices having maximum degree in G is a bipartite graph. Then, there exists a matching in G that matches every vertex with a maximum degree.*

Proof. See [20] for the proof. □

Lemma 2. *Consider a graph G. Suppose that there exists a matching that matches every vertex with a maximum weight, then an MVM also matches every such vertex.*

Proof. The proof is provided in Appendix A. □

Now, we are ready to prove Proposition 1. We provide the detailed proof in Appendix B, and give a sketch of the proof as follows. Consider any time-slot $t \geq 0$. Suppose that the maximum degree is $\Delta \geq 2$ at the beginning of time-slot t. Then, we want to show that under the MVM algorithm, the maximum degree will be at most $\Delta - 2$ at the end of time-slot $t+2$. We use a "1" or "0" to indicate whether the maximum degree decreases by one or not in a time-slot, respectively, and use a sequence of "1"s and "0"s to indicate whether the maximum degree decreases by one or not in a sequence of consecutive time-slots starting from time-slot t. For example, "011" means that the maximum degree does not decrease in time-slot t, but it decreases in both time-slots $t+1$ and $t+2$. Now, we consider two cases in time-slot t: i) "0" occurs and ii) "1" occurs. We will focus on Case i), and Case ii) follows similarly.

In **Case i)**, suppose that in time-slot t, the maximum degree does not decrease under the MVM algorithm. Then, we show by contradiction that at the beginning of time-slot $t+1$, the nodes with a maximum degree Δ must form an independent set, and thus, the subgraph induced by these nodes is a bipartite graph. Further, we show that an MVM must match all the nodes with a maximum degree Δ in time-slot $t+1$. This is due to the following: (1) if the subgraph induced by the nodes with a maximum degree is a bipartite graph, then there exists a matching that matches all the nodes with maximum degree (Lemma 1), and (2) if there exists a matching that matches all the nodes with a maximum degree, then an MVM also matches all such nodes (Lemma 2). Hence, the maximum degree becomes $\Delta - 1$ at the beginning of time-slot $t+2$. Next, we show by contradiction that at the beginning of time-slot $t+2$, the subgraph induced by the nodes with a maximum degree $\Delta - 1$ is a bipartite graph. Then, in time-slot $t+2$ an MVM must match all the nodes with a maximum degree $\Delta - 1$, again from Lemmas 1 and 2.

In **Case ii)**, suppose that "1" occurs in time-slot t, i.e., the maximum degree decreases by one. If another "1" occurs in time-slot $t+1$, then we are done. Now, suppose that "0" occurs in time-slot $t+1$. Then, following the same argument as in Case i), we can show that it will be followed by a "1" in time-slot $t+2$.

Combining the two cases, we complete the proof.

Remark: The proof of Proposition 1 follows a similar argument as in the proof of Theorem 2 of [20]. However, there is a key difference: we provide an actual online algorithm – MVM – that achieves Shannon's bound of $\frac{3}{2}\Delta$ [18], while in [20] only an existence proof is given without providing an actual algorithm. To the best of our knowledge, this is also the first time an online algorithm is proved to achieve Shannon's bound. Moreover, we show in the following corollary that the MVM algorithm actually achieves a bound slightly better than Shannon's.

Corollary 1. $T^{MVM} \le \frac{3}{2}\mathcal{X}' - \frac{1}{2}$.

Proof. The proof is straightforward, and is provided in Appendix C. □

IV. CNM Algorithm

From the above analysis for MVM, we make an important observation: in order to guarantee an approximation of $\frac{3}{2}$ for the minimum evacuation time problem, it suffices to focus on scheduling the critical nodes. Hence, in this section, using the insights obtained, we propose a new scheduling algorithm, called the Critical Node Matching (CNM) algorithm, which focuses on scheduling the critical nodes. We will show that the CNM algorithm has a complexity of $O(m\sqrt{n})$, lower than $O(m\sqrt{n}\log n)$ of MVM, while guaranteeing an approximation ratio no greater than $\frac{3}{2}$ as well.

Note that the MVM algorithm also gives a higher priority to the critical nodes, as the node weight is equal to the node degree, and hence, the critical nodes will have a larger weight. However, we will be able to reduce the complexity of the CNM algorithm by assigning bounded integer weights to the nodes. The specific operations of the CNM algorithm are described as follows. Consider any fixed integer $B > 1$. In each time-slot, we assign the weight of a critical node as B_2, and assign the weight of a non-critical node as B_1, where both B_1 and B_2 are positive integers satisfying $0 < B_1 < B_2 \le B$. Note that the exact values of B_1 and B_2 are not important. For example, in Fig. 1a, we assume $B_1 = 1$ and $B_2 = 2$. Hence, the node weights of $\{a, b, c, d, e\}$ under CNM are $\{1, 1, 2, 2, 1\}$, as nodes c and d are the critical nodes. In every time-slot, the CNM algorithm finds an MVM with assigned node weights of B_1 or B_2. Such a matching is called a CNM. Note that we restrict the node weights to be positive to ensure that the resultant matching is maximal. Fig. 1c shows an example of a CNM but not an MVM, while Fig. 1b shows an example of both a CNM and an MVM. As we discussed earlier, the complexity of finding an MVM is $O(m\sqrt{n}\log n)$ in general. However, in the case where the maximum weight is a bounded integer (independent of both m and n), we can find an MVM with a lower complexity of $O(m\sqrt{n})$ [21], [22]. This can be implemented by setting the weight of an edge to the sum of the weight of its two end nodes and finding an MWM with edge weights being one of the following: $2B_1$ (neither end node is critical), $B_1 + B_2$ (exactly one end node is critical), or $2B_2$ (both end nodes are critical).

Next, we state another main result of this paper in Theorem 2. The proof of Theorem 2 follows from Proposition 2, which follows from Lemmas 1 and 3. Similarly, we have a corollary for CNM. We omit all the proofs here, since they all follow the same line of analysis as in the proofs for the MVM algorithm.

Theorem 2. *The CNM algorithm has an approximation ratio no greater than* $\frac{3}{2}$ *for the minimum evacuation time problem.*

Proposition 2. *Suppose the maximum node degree is no smaller than two. Under the CNM algorithm, the maximum node degree decreases by at least two within every three consecutive time-slots.*

Lemma 3. *Consider a graph* G. *Suppose that there exists a matching that matches every vertex with a maximum weight, then a CNM also matches every such vertex.*

Corollary 2. $T^{CNM} \le \frac{3}{2}\mathcal{X}' - \frac{1}{2}$.

Remark: Note that the CNM algorithm is very similar to the MVM algorithm, in a sense that CNM also needs to find an MVM in each time-slot. However, the key difference is that under CNM, nodes only have a weight of B_2 or B_1, depending on whether a node is a critical node or not. This key difference allows CNM to be able to achieve the same provable evacuation time performance with a reduced complexity.

V. Conclusion

In this paper, we showed that the node-based scheduling algorithms, such as MVM and CNM, achieve a better approximation ratio than the known online algorithms that are either link-based or queue-length agnostic (no worse than $\frac{3}{2}$ vs. no better than 2). However, this work is only among the very first attempts of exploring the idea of the node-based approach, and the performance limits of the node-based scheduling algorithms are far from being fully understood.

On one hand, it is well known in the edge coloring literature that achieving any constant-factor approximation ratio better than $\frac{4}{3}$ is NP-hard [8]. For the MVM algorithm, indeed, we can construct a counterexample to show that its approximation ratio is no better (or smaller) than $\frac{4}{3}$ [11]. On the other hand, a (non-online) algorithm that achieves this $\frac{4}{3}$ bound is given in [23], and moreover, if a small additive term is allowed, much better approximation algorithms and asymptotic approximation algorithms with polynomial time can be developed (see [9] and the references therein).

In [11], we made a *conjecture* that the bound of $\frac{4}{3}$ is tight for the MVM algorithm. In this paper, we made progress towards proving the conjecture by showing that the approximation ratio of the MVM algorithm is between $\frac{4}{3}$ and $\frac{3}{2}$. It is interesting to know whether this gap can be closed. This, however, becomes much more difficult because the MVM algorithm does not possess a nice property similar to Proposition 1, when four consecutive time-slots are considered. That is, it is not generally true that under the MVM algorithm, the maximum node degree decreases by at least three within every four consecutive time-slots. Further, it is also interesting and important to know whether the node-based approach can result in scheduling algorithms with a better approximation ratio for the minimum evacuation time problem when a small additive term is allowed.

Finally, in multihop wireless networks with future packet arrivals, the throughput performance of the node-based scheduling algorithms is not well understood, and is worth further investigation.

Appendix A
Proof of Lemma 2

Proof. We first give some additional notations that will be used in the proof. Define a *path* in a graph as a sequence of edges which connect a sequence of distinct vertices. The *length* of a path is the number of edges in that path. For any two matchings M_1 and M_2, we let $M_1 \ominus M_2 = (M_1 - M_2) \cup (M_2 - M_1)$ denote the *symmetric difference*, which is the set of edges in one of the two matchings, but not in both of them. Given a matching M and a path P, we let $M \oplus P \triangleq M - (M \cap P) + (M^c \cap P)$ denote the set of edges obtained by flipping the edges in P. Then, we define the notions of *augmenting path* and *absorbing path* as in [5], [10]. Consider a given matching M and any vertex i unmatched by M. A path P is an *augmenting path* if it satisfies all the following conditions: i) it has an odd-length; ii) its every alternate edge is in M; iii) it starts from vertex i and ends at another unmatched vertex, say j. Note that now $M \oplus P$ is also a matching, and it matches every vertex of P, including all the vertices matched by M, as well as i and j. Thus, we have $w(M \oplus P) = w(M) + w_i + w_j > w(M)$, i.e., matching $M \oplus P$ has a weight strictly larger than that of M. A path P' is an *absorbing path* if it satisfies all the following conditions: i) it has an even-length; ii) its every alternate edge is in M; iii) it starts from vertex i and ends at a matched vertex k that has a smaller weight than i, i.e., $w_i > w_k$. Note that now $M \oplus P'$ is also a matching, and it matches every vertex of P' except for k, including i and all the vertices of M except for k. Thus, we have $w(M \oplus P') = w(M) + w_i - w_k > w(M)$, i.e., matching $M \oplus P'$ has a weight strictly larger than that of M.

Now, suppose that matching M is an MVM, and M^* is a matching that matches every vertex with a maximum weight Δ. We want to show that M also matches every vertex with weight Δ. We prove this by contradiction. Suppose that there exists a vertex i that has a weight of Δ and is not matched by M (but is matched by M^*). Consider the symmetric difference $M \ominus M^*$. A little thought gives a well-known fact that $M \ominus M^*$ consists of connected components being one of the following: 1) an isolated vertex; 2) an even-length cycle whose edges alternate between M and M^*; 3) a path whose edges alternate between M and M^*, with distinct end nodes. Since vertex i is not matched by M, it must be an end node of a path P in $M \ominus M^*$, otherwise vertex i must be matched by both M and M^*. Hence, we only need to consider scenario 3).

In scenario 3) there are two cases: i) path P has an odd-length and ii) path P has an even-length.

In **Case i)**, path P has an odd-length. Since P begins at vertex i that is not matched by M and has an odd-length, it must end at another vertex that is not matched by M either. Thus, path P is an augmenting path, and hence $M \oplus P$ will be a matching with a larger weight than that of M, which contradicts the fact that M is an MVM.

In **Case ii)**, path P has an even-length. Let vertex j be the other end node of P. Clearly, path P begins at i with an edge in M^* and ends at j with an edge in M, and vertex j is not matched by M^*. Thus, vertex j cannot have a weight of Δ, because M^* matches every vertex with a weight of Δ. Note that i has a weight of Δ and thus $w_i > w_j$. Then, path P is an absorbing path, and thus $M \oplus P$ will be a matching with a weight larger than that of M, which contradicts the fact that M is an MVM.

Combining both cases, we complete the proof. □

Appendix B
Proof of Proposition 1

Proof. First, note that the original graph G becomes a multigraph if we view each packet as a multi-edge. By slightly abusing the notation, we still use G to denote the multigraph. Also, note that multigraph G we consider is *loopless*, i.e., there does not exist a multi-edge connecting a vertex to itself, because a node does not transmit packets to itself.

Now, consider any time-slot $t \geq 0$. Suppose that the maximum degree of G is $\Delta \geq 2$ in time-slot t. Let M_i denote the matching found by the MVM algorithm in time-slots $t+i-1$, for $i = 1, 2, 3$, respectively. Let $G' \triangleq G - M_1 - M_2 - M_3$ denote the graph at the end of time-slot $t+2$. Then, we want to show that the maximum degree of G' is no greater than $\Delta - 2$. We use a "1" or "0" to indicate whether the maximum degree decreases by one or not in a time-slot, respectively, and use a sequence of "1"s and "0"s to indicate whether the maximum degree decreases by one or not in a sequence of consecutive time-slots starting from time-slot t. We then consider the following two cases in time-slot t: i) "0" occurs and ii) "1" occurs.

In **Case i)**, suppose that "0" occurs in time-slot t, i.e., the maximum degree does not decrease in time-slot t. We want to show that it will be followed by "11" in time-slots $t+1$ and $t+2$ under the MVM algorithm.

Note that any MVM must be a maximal matching when all the nodes have a positive weight; otherwise, one can obtain another matching with a larger weight by adding more edges to make it maximal. Since M_1 is an MVM, and is thus a maximal matching, the vertices of degree Δ in $G - M_1$ must form an independent set. We prove it by contradiction. Suppose that there exist two adjacent vertices, say i and j, with degree Δ in $G - M_1$. Then, none of the edges touching either i or j was in matching M_1 in time-slot t. This implies that the edge between i and j can be added to matching M_1 in time-slot t, which, however, contradicts the fact that M_1 is a maximal matching. Therefore, the vertices of degree Δ in $G - M_1$ must form an independent set. Clearly, the subgraph induced by these vertices is a bipartite graph. Then, by Lemma 1, there exists a matching that matches all the vertices with degree Δ in $G - M_1$ in time-slot $t+1$. Note that M_2 is an MVM over $G - M_1$. Then, by Lemma 2, matching M_2 also matches all the vertices with degree Δ in $G - M_1$. This means that a "1" occurs in time-slot $t+1$.

Now, let $H \triangleq G - M_1 - M_2$ denote the graph at the beginning of time-slot $t+2$. From the above analysis, we know that the maximum degree of H is $\Delta - 1$. Let $H_{\Delta-1}$ denote the subgraph of H induced by all the vertices having degree $\Delta - 1$

in H. If $H_{\Delta-1}$ is a bipartite graph, then again by Lemmas 1 and 2, following the same argument above, we can show that matching M_3 matches all the vertices with degree $\Delta - 1$ in time-slot $t+2$. Therefore, it remains to show that $H_{\Delta-1}$ is a bipartite graph. We prove this by contradiction. Suppose that $H_{\Delta-1}$ contains an odd cycle, say C. Then, no two adjacent vertices of C were matched by M_2 in time-slot $t+1$. This is true due to the following. Suppose that there exist two adjacent vertices of C, say i and j, were matched by M_2. Since i and j both have a degree of $\Delta-1$ in H, then they both have degree Δ in $G - M_1$. This contradicts what we have shown earlier - the vertices of degree Δ in $G - M_1$ form an independent set, given that i and j are adjacent. Therefore, no two adjacent vertices of C were matched by M_2 in time-slot $t+1$. This, along with the fact that cycle C is of odd size, implies that cycle C must contain two adjacent vertices that were not matched by M_2 in time-slot $t+1$. This further implies that the edge between these two adjacent vertices can be added to M_2, which contradicts the fact that M_2 is a maximal matching. Therefore, $H_{\Delta-1}$ is a bipartite graph.

So far, we have shown that if "0" occurs in time-slot t, it will be followed by "11" in time-slots $t+1$ and $t+2$.

In **Case ii)**, suppose that "1" occurs in time-slot t, i.e., the maximum degree indeed decreases by one in time-slot t. If another "1" occurs in time-slot $t+1$, then we are done. Now, suppose that "0" occurs in time-slot $t+1$. Then, following the same argument as in Case i), we can show that a "1" will occur in time-slot $t+2$.

Combining both cases, we complete the proof. □

Appendix C
Proof of Corollary 1

Proof. We consider the following two cases: i) $\mathcal{X}' = \Delta$ and ii) $\mathcal{X}' \geq \Delta + 1$. In **Case i)**, it implies that in each time-slot, including the very first time-slot, there exists a matching that matches all the vertices with a maximum degree. Hence, by Lemma 2, the MVM algorithm also matches all the vertices with a maximum degree in the first time-slot. Therefore, by applying Theorem 1 to the remaining graph, we have $T^{\text{MVM}} \leq 1 + \frac{3}{2}(\mathcal{X}' - 1) = \frac{3}{2}\mathcal{X}' - \frac{1}{2}$. In **Case ii)**, it follows from Proposition 1 that $T^{\text{MVM}} \leq \frac{3}{2}\Delta \leq \frac{3}{2}(\mathcal{X}' - 1)$. Combining the two cases, we complete the proof. □

Acknowledgment

This work was supported in part by the NSF under Grants CNS-1449860, CNS-1065444, and ECCS-1231461. The authors would like to thank the reviewers for their valuable comments and suggestions. The first author also thanks Prof. Richard Beigel at Temple University for helpful discussions on this work.

References

[1] 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, no. 12, pp. 1936–1948, 1992.

[2] X. Lin, N. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1452–1463, Aug. 2006.

[3] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Foundations and Trends in Networking*, vol. 1, no. 1, pp. 1–144, 2006.

[4] L. Tassiulas and A. Ephremides, "Dynamic scheduling for minimum delay in tandem and parallel constrained queueing models," *Annals of Operations Research*, vol. 48, no. 4, pp. 333–355, 1994.

[5] G. R. Gupta, "Delay efficient control policies for wireless networks," *Ph.D. thesis, Purdue University*, 2009.

[6] B. Hajek and G. Sasaki, "Link scheduling in polynomial time," *IEEE Transactions on Information Theory*, vol. 34, no. 5, pp. 910–917, 1988.

[7] G. Sharma, R. Mazumdar, and N. Shroff, "On the complexity of scheduling in wireless networks," in *Proceedings of the 12th annual international conference on Mobile computing and networking*. ACM New York, NY, USA, 2006, pp. 227–238.

[8] I. Holyer, "The NP-completeness of edge-colouring," *Siam J. Comput*, vol. 10, no. 4, pp. 718–720, 1981.

[9] P. Sanders and D. Steurer, "An asymptotic approximation scheme for multigraph edge coloring," *ACM Transactions on Algorithms*, vol. 4, no. 2, pp. 1–24, 2008.

[10] G. R. Gupta, S. Sanghavi, and N. B. Shroff, "Node weighted scheduling," in *The 11th ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*. ACM, 2009, pp. 97–108.

[11] B. Ji, "Design of Efficient Resource Allocation Algorithms for Wireless Networks: High Throughput, Small Delay, and Low Complexity," *Ph.D. thesis, The Ohio State University*, 2012.

[12] A. Stolyar, "Maxweight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic," *Annals of Applied Probability*, vol. 14, no. 1, pp. 1–53, 2004.

[13] D. Shah and D. Wischik, "Optimal scheduling algorithms for input-queued switches," in *The 25th IEEE International Conference on Computer Communications (INFOCOM)*, 2006.

[14] X. Lin and N. Shroff, "The impact of imperfect scheduling on cross-Layer congestion control in wireless networks," *IEEE/ACM Transactions on Networking*, vol. 14, no. 2, pp. 302–315, 2006.

[15] C. Joo, X. Lin, and N. Shroff, "Greedy Maximal Matching: Performance Limits for Arbitrary Network Graphs Under the Node-exclusive Interference Model," *IEEE Transactions on Automatic Control*, vol. 54, no. 12, pp. 2734–2744, 2009.

[16] A. Mekkittikul and N. McKeown, "A practical scheduling algorithm to achieve 100% throughput in input-queued switches," in *The 17th IEEE International Conference on Computer Communications (INFOCOM)*, 1998, pp. 792–799.

[17] V. Tabatabaee and L. Tassiulas, "MNCM: a critical node matching approach to scheduling for input buffered switches with no speedup," *IEEE/ACM Transactions on Networking*, vol. 17, no. 1, pp. 294–304, 2009.

[18] C. Shannon, "A theorem on coloring the lines of a network," *J. Math. Phys*, vol. 28, pp. 148–151, 1949.

[19] T. H. Spencer and E. W. Mayr, "Node weighted matching," in *Automata, Languages and Programming*. Springer, 1984, pp. 454–464.

[20] R. P. Anstee and J. R. Griggs, "An application of matching theory of edge-colourings," *Discrete Mathematics*, vol. 156, no. 1, pp. 253–256, 1996.

[21] C.-C. Huang and T. Kavitha, "Efficient algorithms for maximum weight matchings in general graphs with small edge weights," in *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2012, pp. 1400–1412.

[22] S. Pettie, "A simple reduction from maximum weight matching to maximum cardinality matching," *Information Processing Letters*, vol. 112, no. 23, pp. 893–898, 2012.

[23] D. Hochbaum, T. Nishizeki, and D. Shmoys, "A better than 'best possible' algorithm to edge color multigraphs," *Journal of Algorithms*, vol. 7, no. 1, pp. 79–104, 1986.