Throughput Characterization of Node-based Scheduling in Multihop Wireless Networks: A Novel Application of the Gallai-Edmonds Structure Theorem

Bo Ji and Yu Sang
Dept. of Computer and Information Sciences
Temple University
Philadelphia, PA 19122
{boji, yu.sang}@temple.edu

ABSTRACT

Maximum Vertex-weighted Matching (MVM) is an important link scheduling algorithm for multihop wireless networks. Under certain assumptions, it has been shown that if the underlying network graph is bipartite, MVM not only maximizes the throughput in settings with continuous packet arrivals, but also minimizes the evacuation time (i.e., time to drain all the initial packets) in settings without future packet arrivals. Further, even if the network graph is arbitrary, MVM achieves the best known performance guarantee for the evacuation time among existing online link scheduling algorithms. Also, it empirically exhibits close-to-optimal throughput performance and good delay performance. However, in an arbitrary network graph the throughput performance of MVM has not been well understood. To that end, in this paper we aim to carry out a systematic study of the throughput performance of MVM, assuming *single-hop* flows and the *node-exclusive* interference model. Inspired by the celebrated *Gallai-Edmonds structure theorem*, we introduce a novel topological notion, called the *Gallai-Edmonds decomposition factor*, and rigorously prove that the *efficiency ratio* of MVM is no smaller than the Gallai-Edmonds decomposition factor of the network graph. Further, we show that if the smallest size of an odd cycle in a graph is $2m+1$ for a positive integer m, then the Gallai-Edmonds decomposition factor is equal to $2m/(2m+1)$. This implies that the Gallai-Edmonds decomposition factor is at least $2/3$ for an arbitrary graph and is equal to 1 for bipartite graphs. Having these results, the throughput performance of MVM can be well characterized.

CCS Concepts

•**Networks → Network control algorithms; Network performance analysis; Wireless mesh networks;**

MobiHoc'16, July 04-08, 2016, Paderborn, Germany

DOI: http://dx.doi.org/10.1145/2942358.2942388

Keywords

Throughput; Node-based Scheduling; MVM; Wireless Networks; Gallai-Edmonds Structure Theorem

1. INTRODUCTION

Link scheduling is one of the most critical functionalities in multihop wireless networks. The design of efficient wireless scheduling algorithms is very challenging and has been extensively studied in the literature (e.g., see [7,18] and references therein). Among several metrics that are commonly used for evaluating the scheduling performance, the *throughput* and the *evacuation time* are two of the most important ones [8,9,11,12]. In settings *with continuous packet arrivals*, the throughput is measured by the set of traffic loads under which the network can be stabilized, while in settings *without future arrivals*, the evacuation time is defined as the time for draining all the initial data packets in the network.

Ideally, we wish to have a scheduling algorithm that both maximizes the throughput and minimizes the evacuation time in respective settings. However, these two objectives stated in different settings could lead to conflicting scheduling decisions [8,9]. Therefore, it is generally very challenging to design scheduling algorithms that can achieve provably good performance in both dimensions at the same time.

Maximum Vertex-weighted Matching (MVM) is an important scheduling algorithm, which in each scheduling cycle selects a schedule that maximizes the sum of the loads of the scheduled nodes (see Section 5.1 for the detailed description). MVM has recently aroused increasing interests due to its good performance. First, under the assumption of single-hop flows and the node-exclusive interference model [10, 18, 22], it has been shown in [8, 9] that MVM is both throughput-optimal and evacuation-time-optimal if the underlying network graph is bipartite. Second, even if the network graph is arbitrary, MVM achieves the best known guarantee for the evacuation time among existing online scheduling algorithms. Specifically, it has been shown in [12] that MVM has an approximation ratio[1] no greater than $3/2$ for the evacuation time. Finally, extensive numerical studies show that in various scenarios (with different network graphs, arrival processes, and etc.) [8, 9, 11], MVM

[1]Under the node-exclusive interference model, the evacuation time problem can be mapped to the well-known edge coloring problem that is generally NP-hard (e.g., see [12]). The approximation ratio is the worst-case ratio of the evacuation time of an algorithm to the minimum evacuation time.

not only empirically achieves close-to-optimal throughput performance, but also exhibits good delay performance.

However, the throughput performance of MVM has not been well understood. To that end, in this paper we aim to carry out a *systematic study of the throughput performance of MVM*. Throughout the paper, we assume *single-hop* traffic flows (that traverse only one link) and the *node-exclusive* interference model [10, 18, 22], under which links sharing a common node cannot make simultaneous data transmissions. We summarize our key contributions as follows.

First, we introduce a novel topological notion, called the *Gallai-Edmonds decomposition factor* (Definition 5), which is inspired by the celebrated Gallai-Edmonds structure theorem in graph theory and the associated decomposition structure. *To the best of our knowledge, this paper, for the first time, develops such a notion that establishes an interesting connection between the Gallai-Edmonds structure theorem and the throughput performance of a scheduling algorithm in multihop wireless networks.*

Second, we show that *the Gallai-Edmonds decomposition factor is fully determined by the smallest size of an odd cycle in a graph*. Specifically, we prove that if the smallest size of an odd cycle in a graph is $2m+1$, where m is a positive integer, then the Gallai-Edmonds decomposition factor is equal to $2m/(2m+1)$ (Theorem 3). This further implies that the Gallai-Edmonds decomposition factor is no smaller than $2/3$ for an arbitrary graph and is equal to 1 for bipartite graphs (Corollary 1).

Third, we prove several interesting and important properties of MVM (Propositions 1 and 2). Using these key properties, we rigorously prove that *the efficiency ratio of MVM (i.e., the worst-case ratio of the throughput of MVM to that of a throughput-optimal algorithm; see Definition 4) is no smaller than the Gallai-Edmonds decomposition factor of the underlying network graph* (Theorem 4).

Having these results, the throughput performance of MVM can be well characterized. Together with the result on the evacuation time performance of MVM [12], our findings show that among existing online scheduling algorithms, MVM achieves the most balanced performance guarantees in both dimensions of throughput and evacuation time. Not only do these results substantially improve our understanding of the performance of MVM, but also are instrumental in providing important guidelines for the design of wireless networks when MVM-type of algorithms are employed.

The remainder of this paper is organized as follows. We first discuss related work and present our system model in Sections 2 and 3, respectively. In Section 4, we review the Gallai-Edmonds structure theorem and introduce the Gallai-Edmonds decomposition factor. Then, in Section 5 we describe the MVM algorithm and prove some key properties of MVM, and in Section 6 we show that the efficiency ratio of MVM is no smaller than the Gallai-Edmonds decomposition factor of the underlying network graph. Finally, we make concluding remarks in Section 7.

2. RELATED WORK

As mentioned in the introduction, the problem of designing efficient scheduling algorithms for multihop wireless networks has been extensively studied. There has been a significant body of related work on this topic since the seminal work by Tassiulas and Ephremides [25]. In the following discussion, we will focus on prior work that is most relevant to this paper. For more general discussions on this topic, the interested reader is referred to [7, 18] and references therein.

Since the seminal work of [25], it has been shown that the MaxWeight algorithm and its variants are throughput-optimal in very general settings. However, MaxWeight-type of algorithms generally have a high complexity and require a centralized controller. To that end, several lower-complexity and/or distributed approximation algorithms have been developed, including the greedy algorithms (e.g., [14, 17, 26]), random access algorithms (e.g., [15, 16]), and those based on the CSMA (Carrier Sensing Multiple Access) techniques (e.g., [13, 20, 21]), which have been shown to be throughput-efficient. However, under the node-exclusive interference model none of them can achieve an approximation ratio smaller than 2 for the evacuation time [8, 9].

In stark contrast to the aforementioned algorithms that make scheduling decisions based on the link loads or are load-agnostic, another class of algorithms take a node-based approach and make decisions based on the node loads [8, 9, 11, 12, 19, 24]. The node-based approach respects the fact that nodes and odd cycles with maximum load are the bottlenecks for achieving small evacuation time, and typically leads to better evacuation time performance [11]. Specifically, it has been shown in [8, 9] that a class of node-based algorithms, including MVM, are both throughput-optimal and evacuation-time-optimal in input-queued switches[2]. Further, the authors of [12] focus on the evacuation time performance of MVM in arbitrary network graphs and show that MVM achieves an approximation ratio no greater than $3/2$. However, the throughput performance of MVM in an arbitrary network graph has not been well understood, which is the focus of this paper. In [11], two node-based service-balanced scheduling algorithms are proposed, which have been shown to achieve an efficiency ratio no smaller than $2/3$ for the throughput and an approximation ratio no greater than $3/2$ for the evacuation time. However, the way these algorithms are designed renders it difficult to pursue a fine-grained performance characterization when the network graph belongs to different classes that possess structural properties, while in this paper, the throughput performance of MVM can be well characterized through a novel application of the Gallai-Edmonds structure theorem and the associated decomposition structure.

Finally, it is worth noting that in [14] another topological notion called the *local-pooling factor* is proposed to characterize the efficiency ratio of the *Greedy Maximal Matching (GMM)* algorithm. However, there are key differences between [14] and this paper. First, the efficiency ratio of GMM in an arbitrary network graph is lower bounded by $1/2$, which is smaller than $2/3$ for MVM as we show in this paper. Second, the local-pooling factor is generally harder to estimate, while the Gallai-Edmonds decomposition factor introduced in this paper is fully determined by the smallest size of an odd cycle in a graph. Moreover, MVM achieves a better approximation ratio for the evacuation time than GMM (3/2 vs. 2).

3. SYSTEM MODEL

We consider a multihop wireless network and describe it as an undirected, simple graph $G = (V, E)$, where V is the set

[2] The results also apply to multihop wireless networks of which the underlying network graph is bipartite.

of vertices and E is the set of edges. A vertex corresponds to a wireless node that can transmit and receive data packets; an edge corresponds to a wireless link between two nodes[3]. We assume a time-slotted system with one single frequency channel. A time-slot is denoted by $n \in \{0, 1, 2, \dots\}$. For ease of presentation, we assume unit link capacities, i.e., each link can transmit at most one packet per time-slot. However, it is easy to extend the analysis and results to more general models with heterogeneous link capacities. We assume the *node-exclusive* interference model [10,18,22], under which links sharing a common node cannot be activated at the same time. Hence, a feasible schedule corresponds to a *matching*, denoted by M, in the underlying network graph.

In this paper, we focus on the link scheduling problem and assume that all the flows are single-hop (i.e., their packets traverse only one link before leaving the network). Let $\hat{A}_l(n)$ denote the cumulative number of packet arrivals at link $l \in E$ up to (and including) time-slot n. We assume that the arrival processes $\{\hat{A}_l(n), n = 0, 1, 2, \dots\}$ satisfy the strong law of large numbers. That is, with probability one, the following is satisfied for all $l \in E$: $\lim_{n\to\infty} \frac{\hat{A}_l(n)}{n} = \hat{\lambda}_l$, where $\hat{\lambda}_l$ is the mean arrival rate of link l. Let $\hat{\lambda} \triangleq [\hat{\lambda}_l : l \in E]$ be the arrival rate vector. Let $\hat{D}_l(n)$ denote the cumulative number of packet departures at link l up to time-slot n, and let $\hat{Q}_l(n)$ denote the number of packets waiting to be transmitted over link l (or the queue length of link l) at the beginning of time-slot n. We assume that there are a finite number of packets in the system at the beginning of time-slot 0. Also, we assume that packets arrive (respectively, depart) at the beginning (respectively, end) of each time-slot. By convention, we set $\hat{A}_l(0) = \hat{D}_l(-1) = 0$. Then, the queueing dynamics of link l is given by

$$\hat{Q}_l(n) = \hat{Q}_l(0) + \hat{A}_l(n) - \hat{D}_l(n-1). \tag{1}$$

Since the scheduling algorithm of interest (i.e., MVM) takes a node-based approach, we introduce some additional notations associated with the nodes. First, let $L(i)$ denote the set of links that are incident to node $i \in V$, i.e., $L(i) = \{l \in E \mid \text{node } i \text{ is an end node of link } l\}$. Then, let $Q_i(n)$ denote the workload (i.e., number of packets to transmit or receive) at node $i \in V$ at the beginning of time-slot n, i.e., $Q_i(n) \triangleq \sum_{l\in L(i)} \hat{Q}_l(n)$. Similarly, let $A_i(n)$ and $D_i(n)$ denote the cumulative workload arrivals and departures at node i up to time-slot n, respectively, i.e., $A_i(n) = \sum_{l\in L(i)} \hat{A}_l(n)$ and $D_i(n) = \sum_{l\in L(i)} \hat{D}_l(n)$. Hence, the queueing dynamics of node i is given by

$$Q_i(n) = Q_i(0) + A_i(n) - D_i(n-1). \tag{2}$$

Let $\lambda \triangleq [\lambda_i : i \in V]$ denote the node arrival rate vector, where $\lambda_i = \sum_{l\in L(i)} \hat{\lambda}_l$ is the mean arrival rate of node i.

Let $\mathcal{M}$ denote the set of matchings in G. By slightly abusing the notations, for matching $M \in \mathcal{M}$, let $M_l = 1$ if link l is included in M, and $M_l = 0$ otherwise. In each time-slot, a scheduling algorithm will select a matching in $\mathcal{M}$ as a feasible schedule. Let $H_M(n)$ be the number of time-slots in which matching M is selected as a schedule up to time-slot n. Hence, we have the following equations:

$$D_i(n) = \sum_{M\in\mathcal{M}} \sum_{\tau=1}^{n} \sum_{l\in L(i)} M_l(H_M(\tau) - H_M(\tau - 1)), \tag{3}$$

$$\sum_{M\in\mathcal{M}} H_M(n) = n. \tag{4}$$

Next, we give the definition of network stability.

Definition 1. *The network is* ***rate stable*** *under arrival rate vector* $\hat{\lambda} = \{\hat{\lambda}_l : l \in E\}$ *if with probability one, the following is satisfied for all* $l \in E$*:*

$$\lim_{n\to\infty} \frac{\hat{D}_l(n)}{n} = \hat{\lambda}_l. \tag{5}$$

Note that (5) is equivalent to $\lim_{n\to\infty} \frac{D_i(n)}{n} = \lambda_i$ for all $i \in V$. Rate stability is a weak version of stability and only implies that the departure rate is equal to the arrival rate [3]. We focus on rate stability for ease of presenting our main ideas only. Our analysis follows similarly for stronger versions of stability (e.g., positive recurrence type of stability) if stronger assumptions on the arrival processes are made [2].

We present more definitions as follows.

Definition 2. *The* ***throughput region*** *(or stability region) of a scheduling algorithm, denoted by* Λ*, is defined as the set of arrival rate vectors* λ *for which the network is rate stable under this scheduling algorithm.*

Definition 3. *The* ***optimal throughput region****, denoted by* Λ^**, is defined as the union of the throughput regions of all possible scheduling algorithms.*

Definition 4. *The* ***efficiency ratio*** *of a scheduling algorithm, denoted by* γ^**, is defined as the largest fraction of the optimal throughput region* Λ^* *that is contained in the throughput region* Λ *of this scheduling algorithm, i.e.,*

$$\gamma^* \triangleq \sup\{\gamma \mid \gamma\Lambda^* \subseteq \Lambda\}. \tag{6}$$

Throughout the paper, we will use the efficiency ratio to measure the throughput performance of MVM. Clearly, we have $\gamma^* \in [0, 1]$ for any scheduling algorithm. Particularly, a scheduling algorithm is said to be *throughput-optimal* or achieve *throughput optimality* if its efficiency ratio is equal to 1. To assist our analysis, we define the following region:

$$\Psi \triangleq \{\lambda \mid \lambda_i \le 1 \text{ for all } i \in V\}. \tag{7}$$

Note that Ψ is an outer bound of the optimal throughput region (i.e., $\Lambda^* \subseteq \Psi$), since under any scheduling algorithm, at most one packet can be transmitted to or from a node.

4. GALLAI-EDMONDS STRUCTURE THEOREM AND GALLAI-EDMONDS DECOMPOSITION FACTOR

In this section, we will introduce a new topological notion of a graph, called the *Gallai-Edmonds decomposition factor*, which is inspired by the well-known *Gallai-Edmonds structure theorem* and the associated decomposition structure. This notion is novel in the sense that it can be used to characterize the throughput performance of MVM. Specifically, we will later show that the efficiency ratio of MVM is no smaller than the Gallai-Edmonds decomposition factor

[3] Throughout the paper, we interchangeably use the terms "vertex" and "node"; similar for "edge" and "link". However, we tend to use the terms "vertex" and "edge" in the context of a graph, and use the terms "node" and "link" in the context of a multihop wireless network.

of the underlying network graph (Section 6). Moreover, the Gallai-Edmonds decomposition factor is fully determined by the smallest size of an odd cycle in a graph. Specifically, we show that if the smallest size of an odd cycle contained in a graph is $2m+1$ for $m \in \{1, 2, \dots\}$, then the Gallai-Edmonds decomposition factor of this graph is equal to $2m/(2m+1)$. This further implies that the Gallai-Edmonds decomposition factor is no smaller than 2/3 for an arbitrary graph and is equal to 1 for bipartite graphs.

4.1 Gallai-Edmonds Structure Theorem

We begin with a review of two important results in graph theory: the *Tutte-Berge formula* and the *Gallai-Edmonds structure theorem*. We first give the following basic definitions. Recall that a matching is a subset of edges without common vertices. A matching M is said to miss a vertex v if v is not matched by M. A *maximal matching* is a matching M with the following property: if an edge l not in M is added to M, then $M \cup \{l\}$ is no longer a matching. A *maximum matching* (also known as *maximum-cardinality matching*) is a matching that contains the largest number of edges. A *perfect matching* (also known as 1-*factor*) is a matching that matches all the vertices of a graph. A *near-perfect matching* is a matching that misses exactly one vertex. A graph is called *factor-critical* if for every vertex of this graph, there exists a near-perfect matching that misses only that vertex. Any factor-critical graph must be connected, non-bipartite, and of odd size [27].

Next, we give some additional notations. Let $|\cdot|$ denote the cardinality of a set. For a graph $G = (V, E)$, we let G_U denote the subgraph induced by a vertex-subset $U \subseteq V$, let $\theta(G)$ denote the size of a maximum matching in graph G, and let $o(G_U)$ denote the number of odd components (i.e., connected components with an odd number of vertices) in G_U. Note that an odd component could consist of exactly one vertex, which is then called an *isolated vertex*.

We now state the Tutte-Berge formula [27] in Theorem 1.

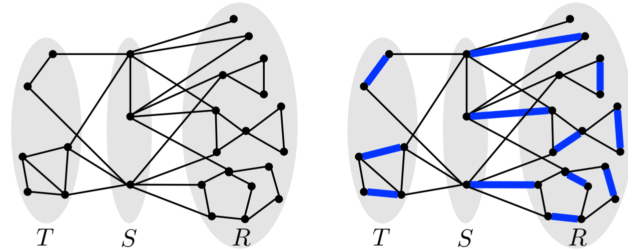
THEOREM 1 (TUTTE-BERGE FORMULA). *The size of a maximum matching in a graph $G = (V, E)$ satisfies:*

$$\theta(G) = \min_{U \subseteq V} \frac{1}{2}(|V| + |U| - o(G_{V \setminus U})). \tag{8}$$

The Tutte-Berge formula is a generalization of Tutte's theorem[4]. While the Tutte-Berge formula provides a characterization of the size of a maximum matching in an arbitrary graph, it does not reveal any structural information of the graph from the matching point of view. The Gallai-Edmonds structure theorem bridges this gap and uncovers interesting and important properties of maximum matchings in an arbitrary graph. In order to state this theorem, we introduce a *canonical decomposition*, which lies in the heart of the Gallai-Edmonds structure theorem. Under this canonical decomposition, the vertex set V can be partitioned into three disjoint vertex-subsets:

$$\begin{aligned} R &\triangleq \{v \in V \mid \text{there exists a maximum matching missing } v\}, \\ S &\triangleq \{v \in V \mid v \text{ is not in } R, \text{ but has a neighbor in } R\}, \\ T &\triangleq V \setminus (R \cup S). \end{aligned} \tag{9}$$

[4]Tutte's (1-factor) theorem gives a necessary and sufficient condition for the existence of a perfect matching in an arbitrary graph. It is a generalization of Hall's marriage theorem from bipartite to arbitrary graphs.



(a) Decomposition Structure (b) A Maximum Matching

Figure 1: An illustration of the Gallai-Edmonds decomposition [1]. (a) presents the decomposition structure, and (b) shows a maximum matching, denoted by thick blue edges.

The above vertex-partition (R, S, T) is also known as the *Gallai-Edmonds decomposition*. The decomposition is well defined even if the graph is not connected. An illustration of the Gallai-Edmonds decomposition is provided in Fig. 1a.

Having defined the above decomposition, we are now ready to restate the Gallai-Edmonds structure theorem [27].

THEOREM 2 (GALLAI-EDMONDS STRUCTURE THEOREM). *Given a graph $G = (V, E)$, let (R, S, T) be the decomposition defined in (9). Then, there exist the following properties:*

i) all the components in G_R are factor-critical and thus are of odd size;

ii) all the components in G_T have a perfect matching and thus are of even size;

iii) every maximum matching in G contains a perfect matching in each component of G_T and a near-perfect matching in each component of G_R, and matches all the vertices in S with vertices in distinct components of G_R (see Fig. 1b for an example of a maximum matching);

iv) choosing $U = S$ achieves the minimum on the right side of the Tutte-Berge formula (Eq. (8)), i.e., the size of a maximum matching is $\theta(G) = \frac{1}{2}(|V| + |S| - o(G_{V \setminus S})) = \frac{1}{2}(|V| + |S| - o(G_R))$.

It is remarkable that the Gallai-Edmonds structure theorem is the foundation of the first polynomial-time algorithm (i.e., the famous blossom algorithm developed by Edmonds [6]) for computing a maximum matching in an arbitrary graph. This theorem reveals an interesting graph structure that leads to many important consequences in graph theory. Nevertheless, as pointed out in [27], this powerful result does not seem to have reached its full potential yet. *To the best of our knowledge, this paper, for the first time, applies this theorem in characterizing the throughput performance of a link scheduling algorithm in multihop wireless networks.* We hope that the results of this paper will serve as an invitation of more efforts towards fully utilizing the power of this beautiful theorem in networking research.

4.2 Gallai-Edmonds Decomposition Factor

We are now ready to introduce the *Gallai-Edmonds Decomposition factor*. We first give some additional notations. For any vertex-subset $U \subseteq V$, let $(R(U), S(U), T(U))$ denote the Gallai-Edmonds decomposition of its induced subgraph G_U. Recall that $G_{R(U)}$ denotes the subgraph induced by

$R(U)$ and that all the components of $G_{R(U)}$ are odd. Let $\mu(G_{R(U)})$ denote the number of components of size larger than one (i.e., components that are not isolated vertices) in $G_{R(U)}$. Then, the Gallai-Edmonds decomposition factor of a graph is defined below.

DEFINITION 5. *For a given graph $G = (V, E)$, consider any vertex-subset $U \subseteq V$ and the Gallai-Edmonds decomposition $(R(U), S(U), T(U))$ of the induced subgraph G_U. The* ***Gallai-Edmonds decomposition factor*** *of graph G, denoted by σ^*, is the supremum of all $\sigma \geq 0$ such that for any $U \subseteq V$, the number of odd components of size larger than one in $G_{R(U)}$ is no greater than $(1-\sigma)|U|$, i.e.,*

$$\sigma^* \triangleq \sup\{\sigma \geq 0 \mid \textit{For any } U \subseteq V, \textit{ the following is satisfied: } \mu(G_{R(U)}) \leq (1-\sigma)|U|\}. \tag{10}$$

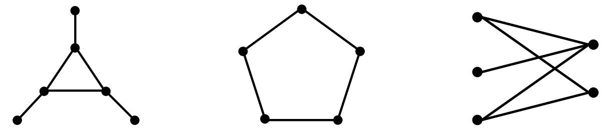
Remark: We highlight that although the Gallai-Edmonds structure theorem is a well-known result in graph theory, *to the best of our knowledge, the Gallai-Edmonds decomposition factor is a new topological notion developed in this paper for the first time*, which depends on the Gallai-Edmonds decomposition of the subgraphs induced by the vertex-subsets. Interestingly, it turns out that the Gallai-Edmonds decomposition factor is closely related to the throughput performance of MVM. Specifically, we will show that the efficiency ratio of MVM is no smaller than the Gallai-Edmonds decomposition factor of the network graph (Theorem 4).

In the sequel, we show that if the smallest size of an odd cycle in a graph is $2m+1$ for $m \in \{1, 2, \dots\}$, then the Gallai-Edmonds decomposition factor of this graph is equal to $2m/(2m+1)$ (Theorem 3). As a consequence, the Gallai-Edmonds decomposition factor is no smaller than $2/3$ for an arbitrary graph and is equal to 1 for bipartite graphs (Corollary 1). We present examples of graphs with different Gallai-Edmonds decomposition factors in Fig. 2.

THEOREM 3. *Consider an arbitrary graph $G = (V, E)$. Suppose the smallest size of an odd cycle in G is $2m+1$ for $m \in \{1, 2, \dots\}$. Then, the Gallai-Edmonds decomposition factor of G is equal to $2m/(2m+1)$, i.e., $\sigma^* = 2m/(2m+1)$.*

PROOF. The proof is straightforward. Suppose the smallest size of an odd cycle in graph G is $2m+1$. We first show that the Gallai-Edmonds decomposition factor is no smaller than $2m/(2m+1)$. Consider any vertex-subset $U \subseteq V$ and the Gallai-Edmonds decomposition $(R(U), S(U), T(U))$ of G_U. Recall that $\mu(G_{R(U)})$ denotes the number of odd components of size larger than one in $G_{R(U)}$. From Property i) of the Gallai-Edmonds structure theorem, we know that all the components in $G_{R(U)}$ are factor-critical and thus must be a non-bipartite graph of odd size [27]. Since any odd cycle contained in graph G has a size no smaller than $2m+1$, then every odd component in $G_{R(U)}$ must contain at least $2m+1$ vertices. Hence, we must have $\mu(G_{R(U)}) \leq |U|/(2m+1)$, which implies that $\mu(G_{R(U)}) \leq (1-\sigma)|U|$ for any $\sigma \leq 2m/(2m+1)$. Therefore, it is implied from Definition 5 that the Gallai-Edmonds decomposition factor must be no smaller than $2m/(2m+1)$, i.e., $\sigma^* \geq 2m/(2m+1)$.

To show that the Gallai-Edmonds decomposition factor is no greater than $2m/(2m+1)$, we only need to consider U as the vertex-subset that consists of all the vertices in an odd cycle of size $2m+1$. In this case, we have $R(U) = U$ and thus $\mu(G_{R(U)}) = |U|/(2m+1)$. Then, we must have $\sigma^* \leq 2m/(2m+1)$, because for any σ greater than $2m/(2m+1)$, the condition in (10) will not be satisfied for U under consideration. This completes the proof. $\square$



(a) $\sigma^* = 2/3$ (b) $\sigma^* = 4/5$ (c) $\sigma^* = 1$

Figure 2: Examples of graphs with different Gallai-Edmonds decomposition factor σ^*. This factor σ^* is fully determined by the smallest size of an odd cycle in the graph.

COROLLARY 1. *The Gallai-Edmonds decomposition factor is no smaller than $2/3$ for an arbitrary graph and is equal to 1 for bipartite graphs.*

PROOF. The proof follows immediately from Theorem 3, since any odd cycle contained in an arbitrary graph has a size no smaller than 3 (i.e., $m = 1$), and a bipartite graph does not contain any odd cycles (i.e., $m = \infty$). $\square$

5. THE MVM ALGORITHM

In this section, we first describe the operations of the Maximum Vertex-weighted Matching (MVM) algorithm (Subsection 5.1), and then prove several interesting properties of MVM (Subsection 5.2), which will play a key role in analyzing the throughput performance of the MVM algorithm.

5.1 Algorithm Description

For a graph $G = (V, E)$, we assume that each vertex $i \in V$ is assigned a positive weight, denoted by $w_i > 0$. We will later describe how to assign the vertex weights. Let $V(M)$ denote the set of vertices matched by matching M. Then, the weight of a matching M is defined as the sum of the weights of all the vertices matched by M, i.e., $w(M) = \sum_{i \in V(M)} w_i$. A matching M^* is called a *Maximum Vertex-weighted Matching (MVM)* if M^* achieves the largest weight over all the matchings in G, i.e., $w(M^*) \geq w(M)$ for any matching M in G. It has been shown in [23] that an MVM can be found in $O(|E|\,|V|^{1/2} \log |V|)$-time, lower than $O(|E|\,|V|)$-complexity of computing its edge-weighted counterpart – Maximum Weighted Matching (MWM).

Now, consider the multihop wireless network we described in Section 3. Recall that $Q_i(n)$ is the queue length of node i at the beginning of time-slot n. In each time-slot n, we assign the weight of a node i as its queue length, i.e., $w_i = Q_i(n)$ for all $i \in V$. Based on the assigned node weights, we compute an MVM. The links of this matching will be activated to transmit packets. We assume that if a link has a zero queue length, then the corresponding edge will not be considered when computing an MVM in that time-slot.

5.2 Properties of MVM

The MVM algorithm and its variants have been studied in several prior work in the literature (e.g., [8, 9, 11, 12, 19, 23]), which shows that MVM exhibits interesting and useful properties. In this section, we present and prove some new properties of MVM, which will be the key to characterizing the efficiency ratio of the MVM algorithm.

Consider a graph $G = (V, E)$ with vertex weights w_i's. Let $\mathcal{H}_k$ denote the set of the k strictly heaviest vertices for $k \in \{1, 2, \ldots, |V|\}$, i.e., $w_i > w_j$ for any $i \in \mathcal{H}_k$ and any $j \in V \backslash \mathcal{H}_k$. We show in Proposition 1 that an MVM maximizes the number of matched vertices in set $\mathcal{H}_k$.

PROPOSITION 1. *For any $k \in \{1, 2, \ldots, |V|\}$ such that set $\mathcal{H}_k$ exists, an MVM, denoted by M^*, maximizes the number of matched vertices in $\mathcal{H}_k$ over all the matchings, i.e., $|V(M^*) \cap \mathcal{H}_k| \geq |V(M) \cap \mathcal{H}_k|$ for any matching M in G.*

PROOF. We first provide some definitions and notations that will be used in the proof. Define a *path* in a graph as a sequence of connected edges. The *length* of a path is the number of edges on the path. For any two edge-subsets $E_1 \subseteq E$ and $E_2 \subseteq E$, let $E_1 \oplus E_2 \triangleq (E_1 \backslash E_2) \cup (E_2 \backslash E_1)$ denote their *symmetric difference*, i.e., the set of edges that are in one of the edge-subsets, but not in both of them. As in [8, 9], we define *augmenting path* and *absorbing path*. Consider a matching M and a vertex i that is missed by M. A path P is called an *M-augmenting path* if it has the following properties: i) it has an odd length; ii) its every alternate edge is in M; iii) it starts from vertex i and ends at another vertex j missed by M. Similarly, a path P' is called an *M-absorbing path* if it has the following properties: i) it has an even length; ii) its every alternate edge is in M; iii) it starts from vertex i and ends at a vertex r, which is matched by M and has a smaller weight than vertex i, i.e., $w_r < w_i$. It is easy to verify that both $M \oplus P$ and $M \oplus P'$ are also a matching and have a weight strictly larger than that of the original matching M because $w(M \oplus P) = w(M) + w_i + w_j > w(M)$ and $w(M \oplus P') = w(M) + w_i - w_r > w(M)$.

We proceed the proof by contradiction. Suppose matching M^* is an MVM and there exists a matching M such that $|V(M) \cap \mathcal{H}_k| > |V(M^*) \cap \mathcal{H}_k|$ for some $k \in \{1, 2, \ldots, |V|\}$. This implies that there exists a vertex in $\mathcal{H}_k$, say i, which is matched only by M but not by M^*. A little thought gives that in the symmetric difference $M^* \oplus M$, vertex i must be one end of a path P whose edges alternate between M and M^*. Note that path P cannot have an odd length due to the following reason. Suppose path P has an odd length, then the other end of P must be another vertex that is missed by M^*, which makes P an M^*-augmenting path. Then, matching $M^* \oplus P$ will have a weight larger than that of M^*, which contradicts the fact that M^* is an MVM. Therefore, path P must have an even length and must end at another vertex, say j, that is only matched by M^* but not by M. Note that vertex j must have a weight no smaller than that of vertex i, i.e., $w_j \geq w_i$, otherwise P would be an M^*-absorbing path, and matching $M^* \oplus P$ will have a weight larger than that of M^*, which again leads to a contradiction. Therefore, vertex j is also in $\mathcal{H}_k$.

Based on the argument above, we know that path P must have an even length; both of its ends – vertices i and j – are in $\mathcal{H}_k$; vertex i is only matched by M, and vertex j is only matched by M^*. Now, we construct a new matching $M_1 = M \oplus P$. By comparing M_1 with M, we make the following two observations:

i) $|V(M_1) \cap \mathcal{H}_k| = |V(M) \cap \mathcal{H}_k|$;

ii) $|V(M_1) \backslash V(M^*) \cap \mathcal{H}_k| = |V(M) \backslash V(M^*) \cap \mathcal{H}_k| - 1$.

The first observation means that M_1 and M match the same number of vertices in $\mathcal{H}_k$. This is true because M_1 and M match exactly the same set of vertices except for i and j, and both vertices i and j are in $\mathcal{H}_k$. The second observation means that compared to M, matching M_1 has one more common vertex matched by M^* in $\mathcal{H}_k$. This is true due to the following reason. First, matching M_1 and M have the same number of common vertices matched by M^* in $\mathcal{H}_k \backslash \{i, j\}$, because M_1 and M match exactly the same set of vertices except for i and j. Second, vertex j is matched by both M_1 and M^*, but is missed by M, and vertex i is matched by M, but is missed by both M_1 and M^*. Hence, matching M_1 has one more common vertex in $\mathcal{H}_k$ (i.e., vertex j) matched by M^* than matching M.

Thanks to these observations, we can repeat the argument above by constructing new matchings $M_2, M_3, \ldots$ in a similar way as M_1 is constructed, until we have a matching M' such that $|V(M') \backslash V(M^*) \cap \mathcal{H}_k| = 0$. This must happen within $|\mathcal{H}_k|$ rounds due to the second observation. This implies that all the vertices in $\mathcal{H}_k$ matched by M' must also be matched by M^*, and thus $|V(M') \cap \mathcal{H}_k| \leq |V(M^*) \cap \mathcal{H}_k|$. Also, we have $|V(M') \cap \mathcal{H}_k| = |V(M) \cap \mathcal{H}_k|$ from the first observation. Hence, we have $|V(M) \cap \mathcal{H}_k| \leq |V(M^*) \cap \mathcal{H}_k|$. This leads to a contradiction and completes the proof. □

Remark: Proposition 1 generalizes Lemma 6 of [23], which states that if there exists a matching that matches all the k heaviest vertices, then an MVM also matches all of them.

Next, in Proposition 2 we present a key property of MVM in multigraphs, where more than one edge, called multi-edge, is allowed between two vertices. Although the discussions in Section 4 are focused on simple graphs, the results there also apply to multigraphs. In particular, the Gallai-Edmonds decomposition factor of a multigraph is equal to that of the corresponding simple graph. Similarly, Proposition 1 also applies to multigraphs. We use $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to denote a multigraph, with $\mathcal{V}$ and $\mathcal{E}$ being the set of vertices and the set of multi-edges, respectively. Let $d_{\mathcal{G}}(i)$ denote the degree of vertex i in $\mathcal{G}$. For a vertex-subset $U \subseteq \mathcal{V}$, let $\mathcal{G}_U$ denote the subgraph of $\mathcal{G}$ induced by U, and let $(R(U), S(U), T(U))$ denote the Gallai-Edmonds decomposition of $\mathcal{G}_U$. We use $\mathcal{G}_{R(U)}$ to denote the subgraph induced by $R(U)$, use $\mathcal{I}(\mathcal{G}_{R(U)})$ to denote the set of isolated vertices in $\mathcal{G}_{R(U)}$, and use $\mu(\mathcal{G}_{R(U)})$ to denote the number of odd components of size larger than one in $\mathcal{G}_{R(U)}$. Note that in any time-slot, the network together with the present packets can be represented by a multigraph, where each multi-edge corresponds to a packet. Then, under the MVM algorithm we specified, the weight of a vertex is equal to the vertex degree, i.e., $w_i = d_{\mathcal{G}}(i)$ for $i \in \mathcal{V}$. Let $\nu(\mathcal{G}) = \frac{|\mathcal{V}|-1}{|\mathcal{V}|}$. We now state Proposition 2 below.

PROPOSITION 2. *Consider a multigraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with maximum vertex degree Δ and a vertex-subset $U \subseteq \mathcal{V}$. Suppose the following conditions are satisfied:*

i) every isolated vertex in $\mathcal{G}_{R(U)}$ has a degree no smaller than a $\nu(\mathcal{G})$-fraction of the maximum vertex degree, i.e., $d_{\mathcal{G}}(i) \geq \nu(\mathcal{G})\Delta$ for any $i \in \mathcal{I}(\mathcal{G}_{R(U)})$;

ii) every vertex in U has a larger degree than a vertex not in U, i.e., $d_{\mathcal{G}}(i) > d_{\mathcal{G}}(j)$ for any $i \in U$ and any $j \notin U$.

Then, an MVM matches at least a σ^*-fraction of the vertices in U, where σ^* is the Gallai-Edmonds decomposition factor.

The proof of Proposition 2 follows immediately from a property of MVM (Proposition 1) and an important result related to the Gallai-Edmonds decomposition (Proposition 3). We present and prove Proposition 3 below.

Proposition 3. *Consider a multigraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with maximum vertex degree Δ and a vertex-subset $U \subseteq \mathcal{V}$. Suppose every isolated vertex in $\mathcal{G}_{R(U)}$ has a degree no smaller than a $\nu(\mathcal{G})$-fraction of the maximum vertex degree, i.e., $d_{\mathcal{G}}(i) \geq \nu(\mathcal{G})\Delta$ for any $i \in \mathcal{I}(\mathcal{G}_{R(U)})$. Then, there exists a matching in $\mathcal{G}$ that matches every vertex in U, except for at most one vertex from each odd component of size larger than one in $\mathcal{G}_{R(U)}$.*

We restate two lemmas (Lemma 6 of [11] and Lemma 3.2.2 of [8]) that will be used in the proof of Proposition 3.

Lemma 1. *Consider a multigraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with maximum vertex degree Δ and a vertex-subset $W \subseteq \mathcal{V}$. Suppose the following conditions are satisfied:*

i) every vertex in W has a degree no smaller than a $\nu(\mathcal{G})$-fraction of the maximum vertex degree, i.e., $d_{\mathcal{G}}(i) \geq \nu(\mathcal{G})\Delta$ for any $i \in W$;

ii) $\mathcal{G}_W$ is bipartite.

Then, there exists a matching M in $\mathcal{G}$ such that every vertex of U is matched by M.

Lemma 2. *Consider a bipartite graph $G = (V_1 \cup V_2, E)$, where (V_1, V_2) is the vertex partition, and E is the set of edges. Let $V_1' \subseteq V_1$ and $V_2' \subseteq V_2$ be a vertex-subset of V_1 and V_2, respectively. Suppose there exist two matchings M_1 and M_2 such that M_1 matches all the vertices in V_1', and M_2 matches all the vertices in V_2'. Then, there exists a matching M that matches all the vertices in $V_1' \cup V_2'$.*

We provide the proof of Proposition 3 below and present an illustration of the key components of the proof in Fig. 3.

Proof of Proposition 3. Consider a vertex-subset $U \subseteq \mathcal{V}$. Note that $\mathcal{G}_U$ denotes the subgraph induced by U. Recall that $(R(U), S(U), T(U))$ denotes the Gallai-Edmonds decomposition of $\mathcal{G}_U$, and $\mathcal{I}(\mathcal{G}_{R(U)})$ denotes the set of isolated vertices in $\mathcal{G}_{R(U)}$. Suppose the condition of Proposition 3 is satisfied: every vertex in $\mathcal{I}(\mathcal{G}_{R(U)})$ has a degree no smaller than a $\nu(\mathcal{G})$-fraction of the maximum vertex degree, i.e., $d_{\mathcal{G}}(i) \geq \nu(\mathcal{G})\Delta$ for any $i \in \mathcal{I}(\mathcal{G}_{R(U)})$. We want to show that there exists a matching in $\mathcal{G}$ that matches every vertex in U, except for at most one vertex from each odd component of size larger than one in $\mathcal{G}_{R(U)}$.

First, we want to show that there exists a matching in $\mathcal{G}$ that matches all the vertices in $\mathcal{I}(\mathcal{G}_{R(U)}) \cup S(U)$. To prove this, there are three steps: 1) we use Lemma 1 to show that there exists a matching M_1 in $\mathcal{G}$ that matches all the vertices in $\mathcal{I}(\mathcal{G}_{R(U)})$; 2) we use Property iii) of Theorem 2 to show that there exists another matching M_2 in $\mathcal{G}$ that matches all the vertices in $S(U)$; 3) we construct a bipartite graph from matchings M_1 and M_2, and then use Lemma 2 to show that there exists a matching M_3 in $\mathcal{G}$ that matches all the vertices in $\mathcal{I}(\mathcal{G}_{R(U)}) \cup S(U)$.

Step 1). First, note that the first condition of Lemma 1 is satisfied for $\mathcal{I}(\mathcal{G}_{R(U)})$, i.e., $d_{\mathcal{G}}(i) \geq \nu(\mathcal{G})\Delta$ for any $i \in \mathcal{I}(\mathcal{G}_{R(U)})$. The second condition of Lemma 1 is also satisfied for $\mathcal{I}(\mathcal{G}_{R(U)})$ due to the fact that all the vertices in $\mathcal{I}(\mathcal{G}_{R(U)})$ are the isolated vertices in $\mathcal{G}_{R(U)}$ and thus the subgraph induced by $\mathcal{I}(\mathcal{G}_{R(U)})$ is bipartite. Hence, it is implied from Lemma 1 that there exists a matching M_1 in $\mathcal{G}$ that matches all the vertices in $\mathcal{I}(\mathcal{G}_{R(U)})$.

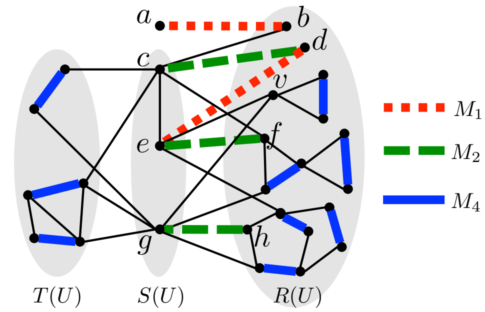


Figure 3: An illustration of the key components in the proof of Proposition 3. Suppose the induced subgraph $\mathcal{G}_U$ is as in Fig. 1a. Then, the set of isolated vertices in $\mathcal{G}_{R(U)}$ is $\mathcal{I}(\mathcal{G}_{R(U)}) = \{b, d\}$. Matchings M_1, M_2, and M_4 are as shown in the above figure. Vertex a is not in U. The constructed bipartite graph $G = (V_1 \cup V_2, E)$ has $V_1 = \{b, d, f, h\}$, $V_2 = \{a, c, e, g\}$, and $E = M_1 \cup M_2$. One option of matching M_3 that matches all the vertices in $\mathcal{I}(\mathcal{G}_{R(U)}) \cup S(U) = \{b, c, d, e, g\}$ is $M_3 = \{(a, b), (c, d), (e, f), (g, h)\}$, where (a, b) denotes the edge between vertices a and b. Then, matching $M = M_3 \cup M_4$ matches every vertex in U, except for at most one vertex from each odd component of size larger than one in $\mathcal{G}_{R(U)}$. In this case, only vertex v is missed by M. (An alternative matching M_3 could be $M_3 = \{(c, b), (e, d), (g, h)\}$. Then, two vertices f and v are missed by the resultant M.)

Step 2). From Property iii) of Theorem 2, we know that every maximum matching in $\mathcal{G}_U$ contains a perfect matching in each component of $\mathcal{G}_{T(U)}$ and a near-perfect matching in each component of $\mathcal{G}_{R(U)}$, and matches all the vertices in $S(U)$ with vertices in distinct components of $\mathcal{G}_{R(U)}$. Let M' be a maximum matching in $\mathcal{G}_U$. Then, the above property is satisfied for M'. Let $M_2 \subseteq M'$ be the set of edges of M' that are incident to vertices in $S(U)$. Clearly, set M_2 is also a matching in $\mathcal{G}$ and matches all the vertices in $S(U)$.

Step 3). Let V_1 be the union of $\mathcal{I}(\mathcal{G}_{R(U)})$ and the vertices in $R(U)$ that are matched by M_2, i.e., $V_1 = \mathcal{I}(\mathcal{G}_{R(U)}) \cup (V(M_2) \cap R(U))$. Let V_2 be the union of $S(U)$ and the vertices not in U that are matched by M_1, i.e., $V_2 = S(U) \cup (V(M_1) \cap (\mathcal{V} \backslash U))$. Let E be the union of matchings M_1 and M_2, i.e., $E = M_1 \cup M_2$. We construct a graph $G = (V_1 \cup V_2, E)$, which is a bipartite graph with (V_1, V_2) being the vertex partition. Then, the conditions in Lemma 2 are satisfied. Specifically, let $V_1' = \mathcal{I}(\mathcal{G}_{R(U)}) \subseteq V_1$ and $V_2' = S(U) \subseteq V_2$. Clearly, M_1 and M_2 are also matchings in G. Moreover, M_1 matches all the vertices in V_1', and M_2 matches all the vertices in V_2'. Therefore, it is implied from Lemma 2 that there exists a matching M_3 in $\mathcal{G}$ that matches all the vertices in $V_1' \cup V_2' = \mathcal{I}(\mathcal{G}_{R(U)}) \cup S(U)$.

Next, we want to show that there exists a matching M in $\mathcal{G}$ that matches all the vertices in U, except for at most one vertex from each odd component of size larger than one in $\mathcal{G}_{R(U)}$. This is a straightforward consequence of the above discussion. To see this, let $M_4 = M' \backslash M_2$ be the set of edges of M' that are not incident to vertices in $S(U)$. It is easy to see that M_4 is also a matching in $\mathcal{G}_U$ (and in $\mathcal{G}$). From Properties iii) of of Theorem 2, we know that M_4 contains a perfect matching in each component of $\mathcal{G}_{T(U)}$ and a near-perfect matching in each component of $\mathcal{G}_{R(U)}$. In other words, matching M_4 matches all the vertices in the union of $T(U)$ and $R(U)$, except for at most one vertex from each of the odd components (including the set of isolated vertices $\mathcal{I}(\mathcal{G}_{R(U)})$) in $\mathcal{G}_{R(U)}$. Let $M = M_3 \cup M_4$. It is easy to see that M is also a matching in $\mathcal{G}$, and that M matches all the vertices in U, except for at most one vertex from each odd component of size larger than one in $\mathcal{G}_{R(U)}$. This completes the proof. □

Remark: Property iv) of the Gallai-Edmonds structure theorem tells us that a maximum matching in $\mathcal{G}_U$ misses $o(\mathcal{G}_{R(U)}) - S(U)$ vertices, each of which is from distinct odd components in $\mathcal{G}_{R(U)}$ and can potentially be an isolated vertex. However, Proposition 3 presents a stronger result under the condition that every isolated vertex in $\mathcal{G}_{R(U)}$ has a degree close to the maximum vertex degree.

We now use Propositions 1 and 3 to prove Proposition 2.

PROOF OF PROPOSITION 2. Consider any $U \subseteq \mathcal{V}$ such that the conditions in Proposition 2 are satisfied. From Proposition 3 and the first condition of Proposition 2, we know that there exists a matching M in $\mathcal{G}$ that matches all the vertices in U, except for at most one vertex from each odd component of size larger than one in $\mathcal{G}_{R(U)}$. From the second condition of Proposition 2, we know that U is the set of the $|U|$ heaviest vertices, i.e., $U = \mathcal{H}_{|U|}$. Then, it follows from Proposition 1 that an MVM matches at least the same number of vertices in U as M does, i.e., an MVM matches at least $|U| - \mu(\mathcal{G}_{R(U)})$ vertices in U. From the definition of Gallai-Edmonds decomposition factor (Definition 5), we have $|U| - \mu(\mathcal{G}_{R(U)}) \geq \sigma^* |U|$. □

Remark: Proposition 1 only shows that MVM maximizes the number of matched vertices among the k heaviest vertices (i.e., $\mathcal{H}_k$), whereas Proposition 2 allows us to quantify the fraction of the matched vertices in such set if certain conditions are satisfied. This nice property of MVM is the key to proving our main result in the next section.

6. THROUGHPUT PERFORMANCE

In this section, we focus on analyzing the throughput performance of the MVM algorithm. A main result of this paper is presented in Theorem 4.

THEOREM 4. *The efficiency ratio of the MVM algorithm is no smaller than the Gallai-Edmonds decomposition factor of the underlying network graph, i.e., $\gamma^* \geq \sigma^*$.*

Our analysis will employ the *fluid limit* techniques [2,4], which not only help simply the analysis through eliminating the unnecessary randomness in the original stochastic network, but also reveal additional properties of the system dynamics that will be useful in the analysis.

We begin by constructing the fluid model as in [2,4]. Recall that $A_i(n)$ and $D_i(n)$ denote the cumulative workload arrivals and departures of node i up to time-slot n, respectively, $Q_i(n)$ denotes the queue length of node i at the beginning of time-slot n, and $H_M(n)$ denotes the cumulative number of time-slots in which matching $M \in \mathcal{M}$ is chosen as a schedule up to time-slot n. For processes $Y = Q, D, H$, we extend their domain from discrete time to continuous time by setting $Y(t) = Y(\lfloor t \rfloor)$. Then, we construct the Markov process $X = \{X(t) : t \geq 0\}$ as $X(t) = (Q(t), D(t), H(t))$, which represents the system dynamics. Following the same argument as in the proof of Theorem 4.1 of [4], we can show that for almost all sample paths and for all positive sequences $x_r \to \infty$, there exists a subsequence x_{r_j} with $x_{r_j} \to \infty$ as $j \to \infty$ such that the following convergence holds *uniformly over compact intervals* of time t:

$$\frac{A_i(x_{r_j} t)}{x_{r_j}} \to \lambda_i t \text{ for all } i \in V, \tag{11}$$

$$\frac{Q_i(x_{r_j} t)}{x_{r_j}} \to q_i(t) \text{ for all } i \in V, \tag{12}$$

$$\frac{D_i(x_{r_j} t)}{x_{r_j}} \to d_i(t) \text{ for all } i \in V, \tag{13}$$

$$\frac{H_M(x_{r_j} t)}{x_{r_j}} \to h_M(t) \text{ for all } M \in \mathcal{M}. \tag{14}$$

Then, the fluid model equations of the system are:

$$q_i(t) = q_i(0) + \lambda_i t - d_i(t) \text{ for all } i \in V, \tag{15}$$

$$\tfrac{d}{dt} d_i(t) = \sum_{M \in \mathcal{M}} \sum_{l \in L(i)} M_l \tfrac{d}{dt} h_M(t) \text{ for all } i \in V, \tag{16}$$

$$\sum_{M \in \mathcal{M}} h_M(t) = t. \tag{17}$$

Any limit (q, d, h) satisfying the above fluid model equations is called a *fluid limit.* Note that functions $q_i(t)$, $d_i(t)$, and $h_i(t)$ are absolutely continuous and are differentiable at almost all times $t \geq 0$, which are called *regular* times. Taking the derivative of both sides of (15) and substituting (16) into it, we obtain that the following is satisfied for all $i \in V$:

$$\tfrac{d}{dt} q_i(t) = \lambda_i - \sum_{M \in \mathcal{M}} \sum_{l \in L(i)} M_l \tfrac{d}{dt} h_M(t). \tag{18}$$

Next, we borrow some definitions and results from [5], which establish an important relationship between the original stochastic network and the fluid model.

DEFINITION 6. *The fluid model is **weakly stable** if for every fluid model solution (q, d, h) with $q(0) = 0$, the following is satisfied: $q(t) = 0$ for all regular times $t \geq 0$.*

LEMMA 3. *A network is rate stable if the associated fluid model is weakly stable.*

Now, we present the proof of Theorem 4.

PROOF OF THEOREM 4. We want to show that under any arrival rate vector λ strictly inside $\sigma^* \Lambda^*$, the MVM algorithm stabilizes the network. Note that λ is also strictly inside $\sigma^* \Psi$ (i.e., $\lambda_i < \sigma^*$ for all $i \in V$) due to $\Lambda^* \subseteq \Psi$. We define $\epsilon \triangleq \min_{i \in V} (\sigma^* - \lambda_i)$. Then, we must have $\epsilon > 0$.

Due to Lemma 3, it suffices to show that the fluid model is weakly stable. We will use the standard Lyapunov analysis by choosing the following Lyapunov function:

$$V(q(t)) = \max_{i \in V} q_i(t). \tag{19}$$

Note that $V(q(t))$ is a non-negative function with $V(0) = 0$. Since we assume that there are a finite number of initial packets in the network, we have $q(0) = 0$ in the fluid limits. In order to show $q(t) = 0$ for all regular times $t \geq 0$, it suffices to show the following: if $V(q(t)) > 0$ for $t > 0$, then $V(q(t))$ has a negative drift and decreases to 0 at least at a given rate. Then, the fluid model is weakly stable according to Definition 6. Therefore, we want to show that for all regular times $t > 0$, the Lyapunov function $V(q(t))$ has a negative drift, i.e., $\frac{d}{dt}V(q(t)) \leq -\epsilon$, whenever $V(q(t)) > 0$.

We first define some useful notions in the fluid limits. For any fixed time t, a node $i \in V$ is called a *critical node* in the fluid limits if it has the largest queue length in the fluid limits, i.e., $q_i(t) = q_{\max} \triangleq \max_{j \in V} q_j(t)$. We use $\mathcal{C}$ to denote the set of critical nodes in the fluid limits at time t, i.e.,

$$\mathcal{C} \triangleq \{i \in V \mid q_i(t) = q_{\max}\}. \tag{20}$$

Further, let $\mathcal{L}$ denote the set of critical nodes that have the largest queue-length derivative at time t, i.e.,

$$\mathcal{L} \triangleq \{i \in \mathcal{C} \mid \tfrac{d}{dt}q_i(t) = \max_{j \in \mathcal{C}} \tfrac{d}{dt}q_j(t)\}. \tag{21}$$

Next, we will use Proposition 2 to show that in every time-slot corresponding to a small time interval around scaled time t, at least a σ^*-fraction of the nodes in $\mathcal{L}$ will be scheduled by the MVM algorithm.

We show that the conditions of Proposition 2 are satisfied for $\mathcal{L}$. Following a similar argument to that in [14] for GMM, we show that Condition ii) of Proposition 2 is satisfied. Since the nodes in $\mathcal{L}$ have the largest queue-length derivative among the nodes in $\mathcal{C}$ at time t, and $q_i(t)$'s are absolutely continuous, there exists a small $\delta_1 > 0$ such that the nodes in $\mathcal{L}$ will have a queue length strictly larger than any other nodes during interval $(t, t+\delta_1)$, i.e., the following is satisfied for any time $\tau \in (t, t+\delta_1)$:

$$\min_{i \in \mathcal{L}} q_i(\tau) > \max_{j \in V \setminus \mathcal{L}} q_j(\tau).$$

This also implies that all the nodes in $\mathcal{L}$ will have a queue length strictly larger than any other nodes in the original stochastic network within all the time-slots corresponding to the scaled time interval $(t, t+\delta_1)$ in the fluid limits.

Following a similar argument to that in [8,9,11], we show that Condition i) of Proposition 2 is satisfied. Let $\hat{q}_{max}$ be the largest queue length of the nodes not in $\mathcal{C}$ in the fluid limits at time t, i.e., $\hat{q}_{max} = \max_{i \notin \mathcal{C}} q_i(t)$. Then, we have $\hat{q}_{max} < q_{max}$. Choosing β small enough such that $\hat{q}_{max} < q_{max} - 3\beta$ and $\beta < \frac{1}{2|V|-1} q_{max}$, we have

$$q_{max} - \beta > \frac{|V|-1}{|V|}(q_{max} + \beta). \tag{22}$$

Recall that $q_i(t)$'s are absolutely continuous. Hence, there exists a small $\delta \in (0, \delta_1]$ such that for all $i \in \mathcal{C}$ and for all times $\tau \in (t, t+\delta)$, the queue lengths in the fluid limits satisfy $q_i(\tau) \in (q_{max} - \beta/2, q_{max} + \beta/2)$. Let x_{r_j} be a positive subsequence for which the convergence to the fluid limits holds. For large enough j, we have $\left|Q_i(x_{r_j}\tau)/x_{r_j} - q_i(\tau)\right| < \beta/2$ for all $\tau \in (t, t+\delta)$. Define a set of consecutive time-slots in the original stochastic system as $N \triangleq \{\lceil x_{r_j} t\rceil, \lceil x_{r_j} t\rceil + 1, \ldots, \lfloor x_{r_j}(t+\delta)\rfloor\}$, which corresponds to the scaled time interval $(t, t+\delta)$ in the fluid limits. Hence, for all $i \in \mathcal{C}$ and for all time-slots $n \in N$, the queue lengths in the original stochastic network satisfy $Q_i(n) \in (x_{r_j}(q_{max} - \beta), x_{r_j}(q_{max} + \beta))$. Then, it is implied from (22) that all the nodes in $\mathcal{C}$ have a queue length no smaller than a $\frac{|V|-1}{|V|}$-fraction of the largest queue length in all time-slots $n \in N$. Condition i) of Proposition 2 is also satisfied for $\mathcal{L}$ due to $\mathcal{L} \subseteq \mathcal{C}$.

Then, it is implied from Proposition 2 that in every time-slot of N in the original stochastic system, at least a σ^*-fraction of the nodes in $\mathcal{L}$ will be scheduled by MVM. According to the pigeonhole principle, there must exist a node $i^* \in \mathcal{L}$ such that i^* has been scheduled by MVM for at least a σ^*-fraction of the time-slots in N, i.e.,

$$\begin{aligned} &\sum_{M \in \mathcal{M}} \sum_{l \in L(i^*)} M_l (H_M(x_{r_j}(t+\delta)) - H_M(x_{r_j} t)) \\ &\geq \sigma^*(\lfloor x_{r_j}(t+\delta)\rfloor - \lceil x_{r_j} t\rceil + 1). \end{aligned} \tag{23}$$

Therefore, the following is satisfied:

$$\begin{aligned} &\sum_{M \in \mathcal{M}} \sum_{l \in L(i^*)} M_l \tfrac{d}{dt} h_M(t) \\ =&\lim_{\delta \to 0} \sum_{M \in \mathcal{M}} \sum_{l \in L(i^*)} M_l \frac{h_M(t+\delta) - h_M(t)}{\delta} \\ \overset{(a)}{=}&\lim_{\delta \to 0} \lim_{j \to \infty} \sum_{M \in \mathcal{M}} \sum_{l \in L(i^*)} \frac{M_l(H_M(x_{r_j}(t+\delta)) - H_M(x_{r_j} t))}{x_{r_j}\delta} \\ \overset{(b)}{\geq}&\lim_{\delta \to 0} \lim_{j \to \infty} \frac{\sigma^*(\lfloor x_{r_j}(t+\delta)\rfloor - \lceil x_{r_j} t\rceil + 1)}{x_{r_j}\delta} \\ =&\sigma^*, \end{aligned} \tag{24}$$

where (a) and (b) are from (14) and (23), respectively.

Then, we have $\frac{d}{dt}q_{i^*}(t) \leq \lambda_{i^*} - \sigma^* \leq -\epsilon$ from (18). Since node i^* has the largest queue-length derivative among nodes in $\mathcal{C}$, we have $\frac{d}{dt}q_i(t) \leq \frac{d}{dt}q_{i^*}(t) \leq -\epsilon$ for all $i \in \mathcal{C}$. Therefore, the fluid model is weakly stable according to Definition 6. This completes the proof by applying Lemma 3. □

THEOREM 5. *Suppose the smallest size of an odd cycle in the underlying network graph is $2m+1$, where m is a positive integer. Then, the efficiency ratio of MVM is no smaller than $2m/(2m+1)$, i.e., $\gamma^* \geq 2m/(2m+1)$. Moreover, the efficiency ratio of MVM is no smaller than $2/3$ in general, i.e., $\gamma^* \geq 2/3$, and MVM is throughput-optimal if the underlying network graph is bipartite.*

PROOF. The proof follows immediately from Theorems 3 and 4 and Corollary 1. □

Remark: Together with the result on the evacuation time performance of MVM [12], Theorem 5 shows that MVM achieves the most balanced performance guarantees in both dimensions of throughput and evacuation time among existing online scheduling algorithms. Also, our new approach offers an alternative means to that of [8,9] for proving throughput optimality of MVM in bipartite graphs.

7. CONCLUSION

In this paper, we carried out a systematic study of the throughput performance of MVM through a novel application of the Gallai-Edmonds structure theorem. We showed that the efficiency ratio of MVM can be well characterized by the Gallai-Edmonds decomposition factor introduced in this paper. Not only do the results of this paper substantially improve our understanding of the throughput performance of MVM, but also provide useful insights for guiding the design of wireless networks when MVM-type of algorithms are employed. We also hope that our findings will shed light on the full utilization of the beautiful Gallai-Edmonds structure theorem in networking research.

Despite the importance of the results we obtained in this paper, in some cases the derived lower bounds of the efficiency ratio could be loose due to the following reasons.

On the one hand, the Gallai-Edmonds decomposition factor based characterization may underestimate the scheduling capability of MVM. For example, suppose the network graph is as in Fig. 2a, which has a Gallai-Edmonds decomposition factor of 2/3. If the vertex-subset $\mathcal{L}$ of interest in the throughput analysis (Theorem 4) is the triangle in the middle, then our analysis gives that at least two vertices of the triangle will be scheduled by MVM in every time-slot of N. However, in some time-slots (when an outer link has a non-zero queue length), all the vertices of the triangle can be scheduled by MVM. Therefore, the efficiency ratio of MVM could be strictly greater than 2/3. On the other hand, the outer bound Ψ used in the throughput analysis (Theorem 4) may be loose when the odd cycles rather than the nodes are the dominating bottlenecks. For example, suppose the network graph is a triangle, which has a Gallai-Edmonds decomposition factor of 2/3. Then, the efficiency ratio of MVM has a lower bound of 2/3. However, it is easy to see that if the link loads are uniform, then $\Lambda^* = 2/3\Psi$, and MVM is actually throughput-optimal. Therefore, the bounds of the efficiency ratio of MVM can potentially be further improved. We leave it to our future work.

8. REFERENCES

[1] J. Akiyama and M. Kano. *Factors and factorizations of graphs: Proof techniques in factor theory*, volume 2031 of *Lecture Notes in Mathematics*. Springer, 2011.

[2] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting. Scheduling in a queuing system with asynchronously varying service rates. *Probability in the Engineering and Informational Sciences*, 18(02):191–217, 2004.

[3] M. Bramson. *Stability of queueing networks*. Springer, 2008.

[4] J. G. Dai. On positive Harris recurrence of multiclass queueing networks: a unified approach via fluid limit models. *The Annals of Applied Probability*, pages 49–77, 1995.

[5] J. G. Dai and B. Prabhakar. The throughput of data switches with and without speedup. In *Proceedings of IEEE INFOCOM*, 2000.

[6] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of mathematics*, 17(3):449–467, 1965.

[7] L. Georgiadis, M. J. Neely, and L. Tassiulas. Resource allocation and cross-layer control in wireless networks. *Foundations and Trends in Networking*, 1(1):1–144, 2006.

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

[9] G. R. Gupta, S. Sanghavi, and N. B. Shroff. Node weighted scheduling. In *Proceedings of ACM SIGMETRICS*, pages 97–108, 2009.

[10] B. Hajek and G. Sasaki. Link scheduling in polynomial time. *IEEE Transactions on Information Theory*, 34(5):910–917, 1988.

[11] B. Ji, G. R. Gupta, and Y. Sang. Node-based Service-Balanced Scheduling for Provably Guaranteed Throughput and Evacuation Time Performance. In *Proceedings of IEEE INFOCOM*, 2016.

[12] B. Ji and J. Wu. Node-based scheduling with provable evacuation time. In *Proceedings of IEEE CISS*, March 2015.

[13] L. Jiang and J. Walrand. A distributed CSMA algorithm for throughput and utility maximization in wireless networks. *IEEE/ACM Transactions on Networking*, 18(3):960–972, 2010.

[14] 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*, 54(12):2734–2744, 2009.

[15] C. Joo and N. B. Shroff. Performance of random access scheduling schemes in multi-hop wireless networks. *IEEE/ACM Transactions on Networking*, 17(5):1481–1493, 2009.

[16] X. Lin and S. B. Rasool. Constant-time distributed scheduling policies for ad hoc wireless networks. *IEEE Transactions on Automatic Control*, 54(2):231–242, 2009.

[17] X. Lin and N. Shroff. The impact of imperfect scheduling on cross-Layer congestion control in wireless networks. *IEEE/ACM Transactions on Networking*, 14(2):302–315, 2006.

[18] X. Lin, N. Shroff, and R. Srikant. A tutorial on cross-layer optimization in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24(8):1452–1463, Aug. 2006.

[19] A. Mekkittikul and N. McKeown. A practical scheduling algorithm to achieve 100% throughput in input-queued switches. In *Proceedings of IEEE INFOCOM*, pages 792–799, 1998.

[20] 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. *IEEE/ACM Transactions on Networking*, 20(3):825–836, 2012.

[21] S. Rajagopalan, D. Shah, and J. Shin. Network adiabatic theorem: an efficient randomized protocol for contention resolution. In *Proceedings of ACM SIGMETRICS*, pages 133–144, 2009.

[22] G. Sharma, R. Mazumdar, and N. Shroff. On the complexity of scheduling in wireless networks. In *Proceedings of ACM Mobicom*, pages 227–238, 2006.

[23] T. H. Spencer and E. W. Mayr. Node weighted matching. In *Automata, Languages and Programming*, pages 454–464. Springer, 1984.

[24] 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*, 17(1):294–304, 2009.

[25] 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*, 37(12):1936–1948, 1992.

[26] X. Wu, R. Srikant, and J. Perkins. Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks. *IEEE Transactions on Mobile Computing*, pages 595–605, 2007.

[27] Q. R. Yu and G. Liu. *Graph factors and matching extensions*. Springer, 2010.