

Throughput-optimal Scheduling in Multi-hop Wireless Networks without Per-flow Information

Bo Ji, Changhee Joo, and Ness B. Shroff

Abstract

In this paper, we consider the problem of link scheduling in multi-hop wireless networks under general interference constraints. Our goal is to design scheduling schemes that do not use per-flow or per-destination information, maintain a single data queue for each link, and exploit only local information, while guaranteeing throughput optimality. Although the celebrated back-pressure algorithm maximizes throughput, it requires per-flow or per-destination information. It is usually difficult to obtain and maintain this type of information, especially in large networks, where there are numerous flows. Also, the back-pressure algorithm maintains a complex data structure at each node, keeps exchanging queue length information among neighboring nodes, and commonly results in poor delay performance. In this paper, we propose scheduling schemes that can circumvent these drawbacks and guarantee throughput optimality. These schemes use either the readily available hop-count information or only the local information for each link. We rigorously analyze the performance of the proposed schemes using fluid limit techniques via an inductive argument and show that they are throughput-optimal. We also conduct simulations to validate our theoretical results in various settings, and show that the proposed schemes can substantially improve the delay performance in most scenarios.

I. INTRODUCTION

Link scheduling is a critical resource allocation functionality in multi-hop wireless networks, and also perhaps the most challenging. The seminal work of [1] introduces a joint adaptive routing and scheduling

B. Ji is with Department of ECE at the Ohio State University. C. Joo is with School of ECE at UNIST, Korea. N. B. Shroff is with Departments of ECE and CSE at the Ohio State University.

Emails: ji@ece.osu.edu, cjoo@unist.ac.kr, shroff@ece.osu.edu.

A preliminary version of this work was presented at IEEE WiOpt 2011.

algorithm, called back-pressure, that has been shown to be throughput-optimal, i.e., it can stabilize the network under any feasible load. This paper focuses on the settings with fixed routes, where the back-pressure algorithm becomes a scheduling algorithm consisting of two components: flow scheduling and link scheduling. The back-pressure algorithm calculates the weight of a link as the product of the link capacity and the maximum “back-pressure” (i.e., the queue length difference between the queues at the transmitting nodes of this link and the next hop link for each flow) among all the flows passing through the link, and solves a MaxWeight problem to activate a set of non-interfering links that have the largest weight sum. The flow with the maximum queue length difference at a link is chosen to transmit packets when the link is activated.

The back-pressure algorithm, although throughput-optimal, needs to solve a MaxWeight problem, which requires centralized operations and is NP-hard in general [2]. To this end, simple scheduling algorithms based on Carrier Sensing Multiple Access (CSMA) [3]–[5] are developed to achieve the optimal throughput in a distributed manner for single-hop traffic, and are later extended to the case of multi-hop traffic [3] leveraging the basic idea of back-pressure.

However, the back-pressure-type of scheduling algorithms (including CSMA for multi-hop traffic) have the following shortcomings: 1) require per-flow or per-destination information, which is usually difficult to obtain and maintain, especially in large networks where there are numerous flows, 2) need to maintain separate queues for each flow or destination at each node, 3) rely on extensive exchange of queue length information among neighboring nodes to calculate link weights, which becomes the major obstacle to their distributed implementation, and 4) may result in poor overall delay performance, as the queue length needs to build up (creating the back-pressure) from a flow destination to its source, which leads to large queues along the route a flow takes [6], [7]. An important question is whether one can circumvent the above drawbacks of the back-pressure-type of algorithms and design throughput-optimal scheduling algorithms that do not require per-flow or per-destination information, maintain a small number of data queues (ideally, a single data queue for each link), exploit only local information when making scheduling decisions, and potentially have good delay performance.

There have been some recent studies (e.g., [6], [8]–[10]) in this direction. A cluster-based back-pressure algorithm that can reduce the number of queues is proposed in [9], where nodes (or routers) are grouped into clusters and each node needs only to maintain separate queues for destinations within its cluster. In [6], the authors propose a back-pressure policy making scheduling decisions in a shadow layer (where counters are used as per-flow shadow queues). Their scheme only needs to maintain a single *First-In First-Out (FIFO)* queue instead of per-flow queues for each link and shows dramatic improvement in the delay

performance. However, their shadow algorithm still requires per-flow information and constant exchange of shadow queue length information among neighboring nodes. The work in [8] proposes to exploit the local queue length information to design throughput-optimal scheduling algorithms. Their approach combined with CSMA algorithms can achieve fully distributed scheduling without any information exchange. Their scheme is based on a two-stage queue structure, where each node maintains two types of data queues: per-flow queues and per-link queues. The two-stage queue structure imposes additional complexity, and is similar to queues with regulators [11], which have been empirically noted to have very large delays. In [10], the authors propose a back-pressure algorithm that integrates the shortest path routing to minimize the average number of hops between each source and destination pair. However, their scheme further increases the number of queues by maintaining a separate queue $\{i, d, k\}$ at each node i for the packets that will be delivered to destination node d within k hops.

Although these algorithms partly alleviate the effect of the aforementioned disadvantages of the traditional back-pressure algorithms, to the best of our knowledge, no work has addressed all the aforementioned four issues. In particular, a critical drawback of the earlier mentioned works is that they require *per-flow or per-destination information* to guarantee throughput optimality. In this paper, we propose a class of throughput-optimal schemes that can remove this per-flow or per-destination information requirement, maintain a single data queue for each link, and remove information exchange. As a by-product, these proposed schemes also improve the delay performance in a variety of scenarios.

The main contributions of our paper are as follows.

First, we propose a scheduling scheme with *per-hop* queues to address the four key issues mentioned earlier. The proposed scheme maintains multiple FIFO queues $Q_{l,k}$ at the transmitting node of each link l . Specifically, any packet whose transmission over link l is the k -th hop forwarding from its source node is stored at queue $Q_{l,k}$. This hop-count information is much easier to obtain and maintain compared to per-flow or per-destination information. For example, hop-count information can be obtained using *Time-To-Live* or *TTL* information in packet headers. Moreover, as mentioned earlier, while the number of flows in a large network is very large, the number of hops is often much smaller. In the Internet, the longest route a flow takes typically has tens of hops¹, while there are billions of users or nodes [14] and thus the number of flows could be extremely large. A shadow algorithm similar to [6] is adopted in our framework, where a shadow queue is associated with each data queue. We consider the MaxWeight algorithm based

¹In the Routing Information Protocol (RIP) [12], the longest route is limited to 15 hops. In general, an upper bound on the length of a route is 255 hops in the Internet, as specified by TTL in the Internet Protocol (IP) [13].

on shadow queue lengths, and show that this per-Hop-Queue-based MaxWeight Scheduler (HQ-MWS) is throughput-optimal using fluid limit techniques via a hop-by-hop inductive argument. For illustration, in this paper, we focus on the centralized MaxWeight-type of policies. However, one can readily extend our approach to a large class of scheduling policies (where fluid limit techniques can be used). For example, combining our approach with the CSMA-based algorithms of [3]–[5], one can completely remove the requirement of queue length information exchange, and develop fully distributed scheduling schemes, under which no information exchange is required. *To the best of our knowledge, this is the first work that develops throughput-optimal scheduling schemes without per-flow or per-destination information in wireless networks with multi-hop traffic.* In addition, we believe that using this type of per-hop queue structure to study the problem of link scheduling is of independent interest.

Second, we have also developed schemes with per-link queues (i.e., a single data queue per link) instead of per-hop queues, extending the idea to per-Link-Queue-based MaxWeight Scheduler (LQ-MWS). We propose two schemes based on LQ-MWS using different queueing disciplines. We first combine it with the *priority* queueing discipline (called PLQ-MWS), where a higher priority is given to the packet that traverses a smaller number of hops, and then prove throughput optimality of PLQ-MWS. It is of independent interest that this type of hop-count-based priority discipline enforces stability. This, however, requires that nodes sort packets according to their hop-count information. We then remove this restriction by combining LQ-MWS with the FIFO queueing discipline (called FLQ-MWS), and prove throughput optimality of FLQ-MWS in networks where flows do not form loops. We further propose fully distributed heuristic algorithms by combining our approach with the CSMA algorithms, and show that the fully distributed CSMA-based algorithms are throughput-optimal under the time-scale separation assumption.

Finally, we show through simulations that the proposed schemes can significantly improve the delay performance in most scenarios. In addition, the schemes with per-link queues (PLQ-MWS and FLQ-MWS) perform well in a wider variety of scenarios, which implies that maintaining per-link queues not only simplifies the data structure, but also can contribute to scheduling efficiency and delay performance.

The remainder of the paper is organized as follows. In Section II, we present a detailed description of our system model. In Section III, we prove throughput optimality of HQ-MWS using fluid limit techniques via a hop-by-hop inductive argument. We extend our ideas to show throughput-optimality of PLQ-MWS and FLQ-MWS in Section IV. Further, we show that our approach combined with the CSMA-based algorithms leads to fully distributed scheduling schemes in Section V. We evaluate different scheduling schemes through simulations in Section VI. Finally, we conclude our paper in Section VII.

II. SYSTEM MODEL

We consider a multi-hop wireless network described by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} denotes the set of nodes and \mathcal{E} denotes the set of links. Nodes are wireless transmitters/receivers and links are wireless channels between two nodes if they can directly communicate with each other. Let $b(l)$ and $e(l)$ denote the transmitting node and receiving node of link $l = (b(l), e(l)) \in \mathcal{E}$, respectively. Note that we distinguish links (i, j) and (j, i) . We assume a time-slotted system with a single frequency channel. Let c_l denote the link capacity of link l , i.e., link l can transmit at most c_l packets during a time slot if none of the links that interfere with l is transmitting at the same time. We assume unit capacity links, i.e., $c_l = 1$ for all $l \in \mathcal{E}$. A flow is a stream of packets from a source node to a destination node. Packets are injected at the source, and traverse multiple links to the destination via multi-hop communications. Let \mathcal{S} denote the set of flows in the network. We assume that each flow s has a single, fixed, and loop-free route that is denoted by $\mathcal{L}(s) = (l_1^s, \dots, l_{|\mathcal{L}(s)|}^s)$, where the route of flow s has $|\mathcal{L}(s)|$ hop-length from the source to the destination, l_k^s denotes the k -th hop link on the route of flow s , and $|\cdot|$ denotes the cardinality of a set. Let $L^{\max} \triangleq \max_s |\mathcal{L}(s)| < \infty$ denote the length of the longest route over all flows. Let $H_{l,k}^s \in \{0, 1\}$ be 1, if link l is the k -th hop link on the route of flow s , and 0, otherwise. Note that the assumption of single route and unit capacity is only for ease of exposition, and one can readily extend the results to more general scenarios with *multiple fixed routes and heterogeneous link capacities*, applying the techniques used in this paper. We also restrict our attention to those links that have flows passing through them. Hence, without loss of generality, we assume that $\sum_s \sum_{k=1}^{|\mathcal{L}(s)|} H_{l,k}^s \geq 1$, for all $l \in \mathcal{E}$.

The interference set of link l is defined as $I(l) \triangleq \{j \in \mathcal{E} \mid \text{link } j \text{ interferes with link } l\}$. We consider a general interference model, where the interference is symmetric, i.e., for any $l, j \in \mathcal{E}$, if $l \in I(j)$, then $j \in I(l)$. A *schedule* is a set of (active or inactive) links, and can be represented by a vector $M \in \{0, 1\}^{|\mathcal{E}|}$, where component M_l is set to 1 if link l is active, and 0 if it is inactive. A schedule M is said to be *feasible* if no two links of M interfere with each other, i.e., $l \notin I(j)$ for all l, j with $M_l = 1$ and $M_j = 1$. Let \mathcal{M} denote the set of all feasible schedules over \mathcal{E} , and let $Co(\mathcal{M})$ denote its convex hull.

Let $F_s(t)$ denote the cumulative number of packet arrivals at the source node of flow s up to time slot t . We assume that packets are of unit length. We assume that each arrival process $F_s(t) - F_s(t-1)$ is an irreducible positive recurrent Markov chain with countable state space, and satisfies the Strong Law

of Large Numbers (SLLN): That is, with probability one,

$$\lim_{t \rightarrow \infty} \frac{F_s(t)}{t} = \lambda_s, \quad (1)$$

for each flow $s \in \mathcal{S}$, where λ_s is the mean arrival rate of flow s . We let $\lambda \triangleq [\lambda_s]$ denote the arrival rate vector. Also, we assume that the arrival processes are mutually independent across flows. (This assumption can be relaxed as in [15].)

As in [15], a stochastic queueing network is said to be *stable*, if it can be described as a discrete-time countable Markov chain and the Markov chain is *stable* in the following sense: The set of positive recurrent states is non-empty, and it contains a finite subset such that with probability one, this subset is reached within finite time from any initial state. When all the states communicate, stability is equivalent to the Markov chain being positive recurrent [16]. We define the *throughput region* of a scheduling policy as the set of arrival rate vectors for which the network is stable under this policy. Further, we define the *optimal throughput region* (or *stability region*) as the union of the throughput regions of all possible scheduling policies, including the offline policies [1]. We denote by Λ^* the optimal throughput region, whereby Λ^* can be represented as

$$\Lambda^* \triangleq \{\lambda \mid \text{for some } \phi \in Co(\mathcal{M}), \sum_s \sum_k H_{l,k}^s \lambda_s \leq \phi_l \text{ for all links } l \in \mathcal{E}\}. \quad (2)$$

An arrival rate vector is strictly inside Λ^* , if the inequalities above are all strict.

Throughout the paper, we let $(z)^+ \triangleq \max(z, 0)$ denote the larger value between z and 0.

III. SCHEDULING WITH PER-HOP QUEUES

In this section, we propose scheduling policies with per-hop queues and shadow algorithm. We will later extend our ideas to developing schemes with per-link queues in Section IV. We describe our scheduling schemes using the centralized MaxWeight algorithm for ease of presentation. Our approach combined with the CSMA algorithms can be extended to develop fully distributed scheduling algorithms in Section V.

A. Queue Structure and Scheduling Algorithm

We start with the description of queue structure, and then specify our scheduling scheme based on per-hop queues and a shadow algorithm. We assume that, at the transmitting node of each link l , a single FIFO data queue $Q_{l,k}$ is maintained for packets whose k -th hop is link l , where $1 \leq k \leq L^{max}$. Such queues are called *per-hop* queues. For notational convenience, we also use $Q_{l,k}(t)$ to denote the queue length of $Q_{l,k}$ at time slot t . Let $\Pi_{l,k}(t)$ denote the service of $Q_{l,k}$ at time slot t , which takes a value

of c_l (i.e., 1 in our setting), if queue $Q_{l,k}$ is active, or 0, otherwise. Let $D_{l,k}(t)$ denote the cumulative number of packet departures from queue $Q_{l,k}$ up to time slot t , and let $\Psi_{l,k}(t) \triangleq D_{l,k}(t) - D_{l,k}(t-1)$ be the number of packet departures from queue $Q_{l,k}$ at time slot t . Since a queue may be empty when it is scheduled, we have $\Psi_{l,k}(t) \leq \Pi_{l,k}(t)$ for all time slots $t \geq 0$. Let $U_{s,k}(t)$ denote the cumulative number of packets transmitted from the $(k-1)$ -st hop to the k -th hop for flow s up to time slot t for $1 \leq k \leq \mathcal{L}(s)$, where we set $U_{s,1}(t) = F_s(t)$. And let $A_{l,k}(t)$ be the cumulative number of aggregate packet arrivals (including both exogenous arrivals and arrivals from the previous hops) at queue $Q_{l,k}$ up to time slot t . Then, we have $A_{l,k}(t) = \sum_s H_{l,k}^s U_{s,k}(t)$, and in particular, $A_{l,1}(t) = \sum_s H_{l,1}^s F_s(t)$. Let $P_{l,k}(t) \triangleq A_{l,k}(t) - A_{l,k}(t-1)$ denote the number of arrivals for queue $Q_{l,k}$ at time slot t . We adopt the convention that $A_{l,k}(0) = 0$ and $D_{l,k}(0) = 0$ for all $l \in \mathcal{E}$ and $1 \leq k \leq L^{\max}$. The queue length evolves as

$$Q_{l,k}(t) = Q_{l,k}(0) + A_{l,k}(t) - D_{l,k}(t). \quad (3)$$

For each data queue $Q_{l,k}$, we maintain a shadow queue $\hat{Q}_{l,k}$, and let $\hat{Q}_{l,k}(t)$ denote its queue length at time slot t . The arrival and departure processes of the shadow queues are controlled as follows. We denote by $\hat{A}_{l,k}(t)$ and $\hat{D}_{l,k}(t)$ its cumulative amount of arrivals and departures up to time slot t , respectively. Also, let $\hat{\Pi}_{l,k}(t)$, $\hat{P}_{l,k}(t) \triangleq \hat{A}_{l,k}(t) - \hat{A}_{l,k}(t-1)$ and $\hat{\Psi}_{l,k}(t) \triangleq \hat{D}_{l,k}(t) - \hat{D}_{l,k}(t-1)$ denote the amount of service, arrivals and departures of queue $\hat{Q}_{l,k}$ at time slot t , respectively. Likewise, we have $\hat{\Psi}_{l,k}(t) \leq \hat{\Pi}_{l,k}(t)$ for $t \geq 0$. We set by convention that, $\hat{A}_{l,k}(0) = 0$ and $\hat{D}_{l,k}(0) = 0$ for all queues $\hat{Q}_{l,k}$. The arrivals for shadow queue $\hat{Q}_{l,k}$ are set to $(1 + \epsilon)$ times the average amount of packet arrivals at data queue $Q_{l,k}$ up to time slot t , i.e.,

$$\hat{P}_{l,k}(t) = (1 + \epsilon) \frac{A_{l,k}(t)}{t}, \quad (4)$$

where $\epsilon > 0$ is a sufficiently small positive number such that $(1 + \epsilon)\lambda$ is also strictly inside Λ^* given that λ is strictly inside Λ^* . Then, the shadow queue length evolves as

$$\hat{Q}_{l,k}(t) = \hat{Q}_{l,k}(0) + \hat{A}_{l,k}(t) - \hat{D}_{l,k}(t). \quad (5)$$

Using these shadow queues, we determine the service of both data queues and shadow queues using the following MaxWeight algorithm.

Per-Hop-Queue-based MaxWeight Scheduler (HQ-MWS): At each time slot t , the scheduler serves data queues $Q_{l,k^*(l)}$ for $l \in M^*$, where

$$k^*(l) \in \operatorname{argmax}_k \hat{Q}_{l,k}(t), \text{ for each link } l \in \mathcal{E}, \quad (6)$$

$$M^* \in \operatorname{argmax}_{M \in \mathcal{M}} \sum_{l \in \mathcal{E}} \hat{Q}_{l,k^*(l)}(t) \cdot M_l. \quad (7)$$

In other words, we set the service of data queue as $\Pi_{l,k}(t) = 1$ if $l \in M^*$ and $k = k^*(l)$, and $\Pi_{l,k}(t) = 0$ otherwise. We also set the service of shadow queues as $\hat{\Pi}_{l,k}(t) = \Pi_{l,k}(t)$ for all l and k .

Remark: The algorithm needs to solve a MaxWeight problem based on the shadow queue lengths, and ties can be broken arbitrarily if there is more than one queue having the largest shadow queue length at a link or there is more than one schedule having the largest weight sum. Note that we have $\Pi_{l,k}(t) = \hat{\Pi}_{l,k}(t)$ under this scheduling scheme, for all links $l \in \mathcal{E}$ and $1 \leq k \leq L^{max}$ and for all time slots $t \geq 0$. Once a schedule M^* is selected, data queues $Q_{l,k^*(l)}$ for links l with $M_l^* = 1$ are activated to transmit packets if they are non-empty, and shadow queues $\hat{Q}_{l,k^*(l)}$ “transmit” shadow packets as well. Note that shadow queues are just counters. The arrival and departure process of a shadow queue is simply an operation of addition and subtraction, respectively.

B. Throughput Optimality

We present the main result of this section as follows.

Proposition 1: HQ-MWS is throughput-optimal, i.e., the network is stable under HQ-MWS for any arrival rate vector λ strictly inside Λ^* .

We prove the stability of the network in the sense that the underlying Markov chain (whose state accounts for both data queues and shadow queues; see Appendix A for the detailed state description) is stable under HQ-MWS, using fluid limit techniques [15], [17]. We provide the proof of Proposition 1 in Appendix A, and discuss the outline of the proof as follows.

Note that the shadow queues serve only single-hop traffic, i.e., after packets in the shadow queues are served, they leave the system without being transmitted to another shadow queue. We also emphasize that the single-hop shadow traffic gets smoothed under the arrival process of (4), and in the fluid limits (which will be formally established in Appendix A), after a finite time, the instantaneous shadow arrival rate is strictly inside the optimal throughput region Λ^* with small enough $\epsilon > 0$. Then, using the standard Lyapunov approach, we can show the stability for the sub-system consisting of shadow queues.

Now, we consider the data queues in the fluid limits starting from the first hop data queue for each link $l \in \mathcal{E}$. Since the arrival process of data queue $Q_{l,1}$ satisfies the SLLN, the instantaneous arrival of shadow queue $\hat{Q}_{l,1}$ will be equal to $(1 + \epsilon) \sum_s H_{l,1}^s \lambda_s$. This implies that the service rate of shadow queue $\hat{Q}_{l,1}$ is no smaller than $(1 + \epsilon) \sum_s H_{l,1}^s \lambda_s$ due to the stability of shadow queues. Then, the service rate of data queue $Q_{l,1}$ is also no smaller than $(1 + \epsilon) \sum_s H_{l,1}^s \lambda_s$ because $\Pi_{l,k}(t) = \hat{\Pi}_{l,k}(t)$ under HQ-MWS. Since the arrival rate of data queue $Q_{l,1}$ is $\sum_s H_{l,1}^s \lambda_s$, the service rate is strictly greater than the arrival rate for $Q_{l,1}$, establishing its stability. Using this as an induction base, we can show the stability of data

queues via a hop-by-hop inductive argument. This immediately implies that the fluid limit model of the joint system is stable under HQ-MWS.

Although our proposed scheme shares similarities with [6], [8], it has important differences. First, in [6], per-flow information is still required by their shadow algorithm. The shadow packets are injected into the network at the sources, and are then “transmitted” to the destinations via multi-hop communications. Their scheme strongly relies on the information exchange of shadow queue lengths to calculate the link weights. In contrast, we take a different approach for constructing the instantaneous arrivals at each shadow queue according to (4) that is based on the average amount of packet arrivals at the corresponding data queue. This method of injecting shadow packets allows us to decompose multi-hop traffic into single-hop traffic for shadow queues and exploit only local information when making scheduling decisions. Second, although the basic idea behind the shadow arrival process of (4) is similar to the service process of the per-flow queues in [8], the scheme in [8] requires per-flow information and relies on a two-stage queue architecture that consists of both per-flow and per-link data queues. In contrast, our scheme needs only per-hop (and not per-flow) information, i.e., the number of hops each packet has traversed, completely removing per-flow information and per-flow queues. This simplification of required information and data structure is critical, due to the fact that the maximum number of hops in a network is usually much smaller than the number of flows in a large network. For example, in the Internet, the longest route a flow takes typically has tens of hops, while there are billions of nodes and thus the number of flows could be extremely large.

Note that the hop-count in our approach is counted from the source. Such per-hop information is easy to obtain (e.g., from *Time-to-Live* or *TTL* information in the Internet and ad hoc networks). At each link, packets with the same hop-count (from the source of each packet to the link) are kept at the same queue, regardless of sources, destinations, and flows, which significantly reduces the number of queues. In Section IV, we extend our approach to the schemes with per-link queues, and further remove even the requirement of per-hop information.

IV. SCHEDULING WITH PER-LINK QUEUES

In the previous section, we show that per-hop-queue-based MaxWeight scheduler (HQ-MWS) achieves optimal throughput performance. In this section, we extend our ideas to developing schemes with *per-link* queues. To elaborate, we show that per-link-queue-based MaxWeight scheduler, when associated with *priority* or *FIFO* queueing discipline, can also achieve throughput optimality.

A. MaxWeight Algorithm with Per-link Queues

We consider a network where each link l has a single data queue Q_l . Let $Q_l(t)$, $A_l(t)$, $D_l(t)$, $\Pi_l(t)$, $\Psi_l(t)$ and $P_l(t)$ denote the queue length, cumulative arrival, cumulative departure, service, departure and arrival at the data queue Q_l , respectively. Also, we maintain a shadow queue \hat{Q}_l associated with each Q_l , and let $\hat{Q}_l(t)$, $\hat{A}_l(t)$, $\hat{D}_l(t)$, $\hat{\Pi}_l(t)$, $\hat{\Psi}_l(t)$ and $\hat{P}_l(t)$ denote the queue length, cumulative arrival, accumulative departure, service, departure and arrival at the shadow queue \hat{Q}_l , respectively. Similar to (4) for per-hop shadow queues, we control the arrivals to the shadow queue \hat{Q}_l as

$$\hat{P}_l(t) = (1 + \epsilon) \frac{A_l(t)}{t}, \quad (8)$$

where $\epsilon > 0$ is a sufficiently small positive number.

Next, we specify the MaxWeight algorithm with per-link queues as follows.

Per-Link-Queue-based MaxWeight Scheduler (LQ-MWS): At each time slot t , the scheduler serves links in M^* (i.e., $\Pi_l(t) = 1$ for $l \in M^*$, and $\Pi_l(t) = 0$ otherwise), where

$$M^* \in \operatorname{argmax}_{M \in \mathcal{M}} \sum_{l \in \mathcal{E}} \hat{Q}_l(t) \cdot M_l.$$

Also, we set the service of shadow queues as $\hat{\Pi}_l(t) = \Pi_l(t)$ for all l .

Similar as in HQ-MWS, the shadow traffic under LQ-MWS gets smoothed due to the shadow arrival assignment of (8), and the instantaneous arrival rate of shadow queues can be shown to be strictly inside the optimal throughput region Λ^* . Hence, we show in Lemma 20 (see Appendix D) that the fluid limit model for the sub-system consisting of shadow queues is stable under LQ-MWS, using the standard Lyapunov approach and following the same line of analysis for HQ-MWS.

B. LQ-MWS with Priority Discipline

We develop a scheduling scheme by combining LQ-MWS with priority queueing discipline, called **PLQ-MWS**. Regarding priority of packets at each per-link queue, we define *hop-class* as follows: A packet has hop-class- k , if the link where the packet is located is the k -th hop from the source of the packet. When a link is activated to transmit packets, packets with a smaller hop-class will be transmitted earlier; and packets with the same hop-class will be transmitted in a FIFO fashion.

Proposition 2: PLQ-MWS is throughput-optimal.

We provide the outline of the proof and refer to Appendix E for the detailed proof. Basically, we follow the line of analysis for HQ-MWS using fluid limit techniques and induction method. Since a link transmits packets according to their priorities (i.e., hop-classes or hop-count from their respective source nodes), we

can view packets with hop-class- k at link l as in a sub-queue $Q_{l,k}$ (similar to the per-hop queues under HQ-MWS). Now, we consider the data queues in the fluid limits. Since the exogenous arrival process satisfies the SLLN, the instantaneous arrival to shadow queue \hat{Q}_l will be at least $(1 + \epsilon) \sum_s H_{l,1}^s \lambda_s$ for each link $l \in \mathcal{E}$. This implies that the service rate of shadow queue \hat{Q}_l is no smaller than $(1 + \epsilon) \sum_s H_{l,1}^s \lambda_s$ due to the stability of the shadow queues (see Lemma 20 in Appendix D). Then, the service rate of sub-queue $Q_{l,1}$ is also no smaller than $(1 + \epsilon) \sum_s H_{l,1}^s \lambda_s$, because: 1) $\Pi_l(t) = \hat{\Pi}_l(t)$ under PLQ-MWS; and 2) the highest priority is given to sub-queue $Q_{l,1}$ when link l is activated to transmit. Since the arrival rate of sub-queue $Q_{l,1}$ is $\sum_s H_{l,1}^s \lambda_s$, the service rate is strictly greater than the arrival rate for sub-queue $Q_{l,1}$, establishing its stability. Similarly, we can show that the hop-class- j sub-queues are stable for all $j \leq k + 1$, given the stability of the hop-class- j' sub-queues for all $j' \leq k$. Therefore, we can show the stability of the data queues via a hop-by-hop inductive argument. This immediately implies that the fluid limit model of the joint system is stable under PLQ-MWS.

We emphasize that a “bad” priority discipline may cause instability (even in wireline networks). See [18], [19] for two simple counterexamples showing that in a two-station network, a static priority discipline that gives a higher priority to customers with a larger hop-count, may result in instability. (Interested readers are also referred to Chapter 3 of [16] for a good summary of the instability results.) The key intuition of these counterexamples is that, by giving a higher priority to packets with a larger hop-count in one station, the priority discipline may impede forwarding packets with a smaller hop-count to the next-hop station, which in turn starves the next-hop station. On the other hand, PLQ-MWS successfully eliminates this type of inefficiency by giving a higher priority to the packets with a smaller hop-count, and continues to push the packets to the following hops.

Note that PLQ-MWS is different from HQ-MWS, although they appear to be similar. HQ-MWS makes scheduling decisions based on the queue length of each per-hop shadow queue. This may result in a waste of service if a per-hop queue is activated but does not have enough packets to transmit, even though the other per-hop queues of the same link have packets. In contrast, PLQ-MWS makes decisions based on the queue length of each per-link shadow queue and allows a link to transmit packets of multiple hop-classes, avoiding such an inefficiency. The performance difference due to this phenomenon will be illustrated through simulations in Section VI. Furthermore, the implementation of PLQ-MWS is easier than HQ-MWS, since PLQ-MWS needs to maintain only one single shadow queue per link.

Another aspect of PLQ-MWS we would like to discuss is about the *hop-count-based priority discipline* in the context of multi-class queueing networks (or wireline networks). In operations research, stability of multi-class queueing networks has been extensively studied in the literature (e.g., see [16] and the

references therein). To the best of our knowledge, however, there is very limited work on the topic of “priority enforces stability” [20]–[22]. In [20], [21], the authors obtained sufficient conditions (based on linear or piecewise linear Lyapunov functions) for the stability of a multiclass fluid network and/or queueing network under priority disciplines. However, to verify these sufficient conditions relies on verifying the feasibility of a set of inequalities, which in general can be very difficult. The most related work to ours is [22]. There, the authors showed that under the condition of “Acyclic Class Transfer”, where customers can switch classes unless there is a loop in class transfers, a simple priority discipline stabilizes the network under the usual traffic condition (i.e., the normalized load is less than one). Their priority discipline gives a higher priority to customers that are closer to their respective sources.

Interestingly, our hop-count-based priority discipline (for wireline networks) is similar to the discipline proposed in [22]. However, there is a major difference in that while [22] studies stability of wireline networks (without link interferences) under the usual traffic condition, we consider stability of wireless networks with interference constraints that impose the (link) scheduling problem, which is much more challenging. In wireless networks, the service rate of each link depends on the underlying scheduling scheme, rather than being fixed as in wireline networks. Hence, the difficulty is to establish the usual traffic condition by designing appropriate wireless scheduling schemes. In this paper, we develop PLQ-MQS scheme and show that the usual traffic condition and then stability can be established via a hop-by-hop inductive argument under the PLQ-MWS scheme.

C. LQ-MWS with FIFO Discipline

In this section, we develop a scheduling scheme, called **FLQ-MWS**, by combining the LQ-MWS algorithm developed in Section IV-A with *FIFO* queueing discipline (instead of priority queueing discipline), and show that this scheme is throughput-optimal if flows do not form loops. We emphasize that FLQ-MWS requires neither per-flow information nor hop-count information.

To begin with, we define a positive integer $r(l)$ as the rank of link $l \in \mathcal{E}$, and call $R(\mathcal{E}) = (r(l), l \in \mathcal{E})$ a ranking of \mathcal{E} . Recall that $\mathcal{L}(s)$ denotes the loop-free route of flow s . In the following, we prove a key property of the network where flows do not form loops, which will be used to prove the main results in this section.

Lemma 3: Consider a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a set of flows \mathcal{S} , where the flows do not form loops. There exists a ranking $R(\mathcal{E})$ such that the following two statements hold:

- 1) For any flow $s \in \mathcal{S}$, the ranks are monotonically increasing when one traverses the links of flow s from l_1^s to $l_{|\mathcal{L}(s)|}^s$, i.e., $r(l_i^s) < r(l_{i+1}^s)$ for all $1 \leq i < |\mathcal{L}(s)|$.

2) The packet arrivals at a link are either exogenous, or forwarded from links with a smaller rank.

We provide the proof of Lemma 3 in Appendix F. Note that such a ranking with the monotone property exists because the flows do not form a loop. In contrast, it is clear that if flows form a loop, then such a ranking does not exist. Two examples of the networks where flows do not form loops are provided in Figs. 5(b) and 5(c), and an example of the network where flows do form a loop is provided in Fig. 5(a). Note that the ranking is only for the purpose of analysis and plays a key role in proving the system stability under FLQ-MWS, while it will not be used in the actual link scheduling algorithm.

Now, we give the main results of this section in the following proposition.

Proposition 4: FLQ-MWS is throughput-optimal in networks where *flows do not form loops*.

We omit the detailed proof and refer to Appendix F. In the following, we provide the outline of the proof. Motivated by Lemma 3, we extend our analysis for HQ-MWS (or PLQ-MWS). Compared to the PLQ-MWS algorithm, there are differences only in the operations with data queues, and the underlying LQ-MWS algorithm remains the same. Thus, the shadow queues will exhibit similar behaviors, and the fluid limit model for the sub-system of shadow queues is stable under FLQ-MWS (see Lemma 20 in Appendix D). Also, note that Lemma 3 implies that given the qualified ranking (without loss of generality, assuming that the smallest rank is 1), the packet arrivals at links with rank 1 are all exogenous, then following a similar argument in the proof of Proposition 1, we can prove the stability of the corresponding data queues by showing that the instantaneous arrival rate is less than the instantaneous service rate. Since Lemma 3 also implies that the packet arrivals at links with rank 2 are either exogenous or from links with rank 1, we can similarly show the stability of links with rank 2. Repeating the above argument, we can prove the stability of all data queues by induction, which completes the proof of Proposition 4.

Corollary 5: FLQ-MWS is throughput-optimal in tree networks.

The above corollary follows immediately from Proposition 4, because a tree network itself does not contain a cycle of links and flows are all loop-free.

V. EXTENSION TO CSMA-BASED DISTRIBUTED ALGORITHMS

In this section, we employ CSMA techniques to develop fully distributed throughput-optimal scheduling schemes for multi-hop traffic. We consider per-link-queue-based schemes combined with the CSMA-based scheduling of [4].

A. Basic Scheduling Algorithm

We start with description of basic scheduling algorithm based on CSMA. As in [4], we divide each time slot t into a *control* slot and a *data* slot, where the control slot is further divided into W mini-slots. The purpose of the control slot is to generate a collision-free transmission schedule $M(t) \in \mathcal{M}$. To this end, the distributed CSMA scheduling selects at each time slot a set of links that form a feasible schedule. Such a schedule is called a *decision* schedule and used to change links' state (between active and inactive). Let $\sigma(t)$ denote a decision schedule at time slot t .

Let $\mathcal{M}_0 \subseteq \mathcal{M}$ denote the set of possible decision schedules under our CSMA-based algorithm. A decision schedule is selected through a randomized procedure, e.g., a decision schedule $\sigma(t) \in \mathcal{M}_0$ is selected with a positive probability $\alpha(\sigma(t))$ satisfying that $\sum_{\sigma(t) \in \mathcal{M}_0} \alpha(\sigma(t)) = 1$. Based on the decision schedule, the schedule for actual data transmission is determined as follows. For each link $l \in \sigma(t)$, if no link in its interfering neighbors $I(l)$ was active at time slot $t - 1$, then the state of link l becomes active with probability p_l (which will be specified later) and inactive with probability $\bar{p}_l = 1 - p_l$ during time slot t . If at least one link in $I(l)$ was active in the previous time slot, then link l remains inactive² in the current data slot. Any link $l' \notin \sigma(t)$ will have its state unchanged from the previous time slot. Since the current state $M(t)$ depends only on the previous state $M(t - 1)$ and the randomly selected decision schedule $\sigma(t)$, the transmission schedule $M(t)$ evolves as a discrete-time Markov chain (DTMC). Our basic scheduling algorithm is very similar to that of [4]. The key difference is that the link activation probability is based on the shadow queue lengths instead of the data queue lengths. We refer the readers to [4] for the detailed operations of the CSMA-based algorithms.

B. Distributed Implementation with Per-link Queues

In this section, we describe our distributed CSMA-based scheduling scheme with per-link queues, called **LQ-CSMA**. The LQ-CSMA algorithm can be combined with priority or FIFO queueing discipline to develop fully distributed scheduling schemes.

We use the system settings and notations of per-link-queue structure as in Section IV. We also control the shadow arrivals as (8). As in [4], we set link activation probability $p_l = \frac{e^{w_l(t)}}{e^{w_l(t)} + 1}$, where $w_l(t)$ is the weight of link l . We begin with defining a class of functions that will be used for weight calculation. As in [4], [23], let \mathcal{B} denote the set of functions $g(\cdot) : [0, \infty] \rightarrow [0, \infty]$ that satisfy the following conditions:

- 1) $g(x)$ is a non-decreasing and continuous function with $\lim_{x \rightarrow \infty} g(x) = \infty$.

²In the previous data slot, link l must be inactive since the schedule must be feasible.

- 2) Given any $M_1 > 0, M_2 > 0$ and $0 < \epsilon < 1$, there exists a $B < \infty$, such that for all $x > B$, we have $(1 - \epsilon)g(x) \leq g(x - M_1) \leq g(x + M_2) \leq (1 + \epsilon)g(x)$.

For example, functions $g(x) = \log(x + 1)$, $g(x) = x^\alpha$ with $\alpha > 0$, and $g(x) = e^{\sqrt{x}}$ belong to \mathcal{B} , while $g(x) = e^x$ does not. Similar to Chapter 4 of [24], to guarantee the existence of the fluid limit, we further define \mathcal{C} as a subset of \mathcal{B} such that $g(0) = 0$, and for any (x_1, \dots, x_n) and (y_1, \dots, y_n) in $[0, \infty]^n$ and for any $\eta \in [0, 1]$,

$$\sum_i g(x_i) \geq \eta \sum_i g(y_i) \Rightarrow \sum_i g(rx_i) \geq \eta \sum_i g(ry_i), \text{ for all } r > 0. \quad (9)$$

For example, $g(x) = x^\alpha$ with $\alpha > 0$ is in \mathcal{C} .

We set the weight of link $l \in \mathcal{E}$ at time slot t as $w_l(t) = g_l(\hat{Q}_l(t))$, where $g_l \in \mathcal{C}$. We highlight the differences from the original CSMA-based scheduling schemes as follows: i) the link weight is calculated by a function in set \mathcal{C} instead of \mathcal{B} . This restriction is necessary to apply the fluid limit techniques; ii) the shadow queue length $\hat{Q}_l(t)$ is used for the weight calculation instead of the data queue length $Q_l(t)$. The following scheduling scheme is an extension of per-link-queue-based scheduling schemes to CSMA-based algorithm.

Per-Link-Queues-and-CSMA-based Scheduling Algorithm (LQ-CSMA):

Let $p_l = \frac{e^{w_l(t)}}{e^{w_l(t)} + 1}$, where $w_l(t) = g_l(\hat{Q}_l(t))$ is an appropriate function of the shadow queue length of link l as shown above. At the beginning of each time slot, each link l randomly selects a backoff time among $\{0, 1, 2, \dots, W - 1\}$, where W denotes the contention window size. Link l will send an INTENT message to announce its decision of attempting channel when this backoff time expires, unless an interfering link in $I(l)$ sent an INTENT message in an earlier mini-slot. The details are shown in Algorithm 1, which is similar to the Q-CSMA algorithm of [4], except that the activation probability p_l is now determined based on the shadow queue lengths.

Remark: The weight function $g_l(\hat{Q}_l(t))$ needs to be appropriately chosen such that the DTMC of the transmission schedules converge faster compared to the dynamics of the link weights. For example³, $g_l(Q_l(t)) = \alpha Q_l(t)$ with a small α is suggested as a heuristic to satisfy the time-scale separation assumption in [3] and $g_l(Q_l(t)) = \log \log(Q_l(t) + e)$ is used in the proof of throughput optimality in [5] to essentially separate the time scales. In addition, it has been reported in [4] that the weight function $g_l(Q_l(t)) = \log(\alpha Q_l(t))$ with a small α gives the best empirical delay performance. In this

³In [3]–[5], the weight function g_l is a function of the queue length $Q_l(t)$ rather than $\hat{Q}_l(t)$.

Algorithm 1 LQ-CSMA (at time slot t)

- 1) Link l selects a random (integer) backoff time B_l uniformly in $[0, W - 1]$ and waits for B_l control mini-slots.
 - 2) IF link l hears an INTENT message from a link in $I(l)$ before the $(B_l + 1)$ -st control mini-slot, l will not be included in $\sigma(t)$ and will not transmit an INTENT message anymore. Link l will set $M_l(t) = M_l(t - 1)$.
 - 3) IF link l does not hear an INTENT message from any link in $I(l)$ before the $(B_l + 1)$ -st control mini-slot, it will send (broadcast) an INTENT message to all links in $I(l)$ at the beginning of the $(B_l + 1)$ -st control mini-slot.
 - If there is a collision (i.e., if there is another link in $I(l)$ transmitting an INTENT message in the same mini-slot), link l will not be included in $\sigma(t)$ and will set $M_l(t) = M_l(t - 1)$.
 - If there is no collision, link l will be included in $\sigma(t)$ and decide its state as follows:
 - if** no links in $I(l)$ were active in the previous data slot **then**
 - $M_l(t) = 1$ with probability p_l , $0 < p_l < 1$;
 - $M_l(t) = 0$ with probability $\bar{p}_l = 1 - p_l$.
 - else**
 - $M_l(t) = 0$.
 - end if**
 - 4) IF $M_l(t) = 1$, link l will transmit a packet in the data slot, and will set $\hat{Q}_l(t) = (\hat{Q}_l(t) - 1)^+$.
-

paper, we make the time-scale separation assumption as in [3], [4] and assume that the DTMC is in the steady state at every time slot.

Applying Lemma 3 of [4], we can show that the transmission schedule $M(t)$ produced by LQ-CSMA is feasible and the decision schedule σ satisfies $\bigcup_{\sigma \in \mathcal{M}_0} \sigma = \mathcal{E}$ when $W \geq 2$. Applying Proposition 1 of [4], we can obtain that the DTMC of the transmission schedules is irreducible and aperiodic (and reversible in this case), and has the following product-form stationary distribution:

$$\mu(M) = \frac{1}{\kappa} \prod_{l \in \mathcal{M}} \frac{p_l}{\bar{p}_l}, \quad (10)$$

$$\kappa = \sum_{M \in \mathcal{M}} \prod_{l \in \mathcal{M}} \frac{p_l}{\bar{p}_l}. \quad (11)$$

Then from Proposition 2 of [4], we can obtain the following lemma.

Lemma 6: If the window size $W \geq 2$, LQ-CSMA has the product-form distribution given by (10). Further, given any ζ and γ , $0 < \zeta, \gamma < 1$, there exists a $Q_B > 0$ such that: at any time slot t , with

probability greater than $1 - \zeta$, LQ-CSMA chooses a schedule $M(t) \in \mathcal{M}$ that satisfies

$$\sum_{l \in \mathcal{E}} w_l(t) \cdot M_l(t) \geq (1 - \gamma) \max_{M \in \mathcal{M}} \sum_{l \in \mathcal{E}} w_l(t) \cdot M_l \quad (12)$$

whenever $\|\hat{Q}(t)\| > Q_B$.

We omit the proof and refer interested readers to [4] (Lemma 3, Propositions 1 and 2) for details.

Note that we have (80) since Lemma 19 also holds under LQ-CSMA. Applying Lemma 6 and following the same line of analysis for the proof of Lemma 15, we can easily show that the sub-system of shadow queues \hat{q} is stable under LQ-CSMA in the fluid limit model.

Lemma 7: Given any ζ and γ , $0 < \theta, \gamma < 1$, with probability greater than $1 - \theta$, the sub-system of shadow queues \hat{q} operating under LQ-CSMA satisfies that: For any $\zeta > 0$, there exists a finite $T_4 > 0$ such that, for any fluid model solution with $\|\hat{q}(0)\| \leq 1$, we have

$$\|\hat{q}(t)\| \leq \zeta, \text{ for all } t \geq T_4, \quad (13)$$

for any arrival rate vector strictly inside $(1 - \gamma)\Lambda^*$.

The proof is provided in Appendix I.

The LQ-CSMA algorithm combined with priority queueing discipline and FIFO queueing discipline is called **PLQ-CSMA** and **FLQ-CSMA**, respectively. We present the main results of this section as follows.

Proposition 8: PLQ-CSMA is throughput-optimal.

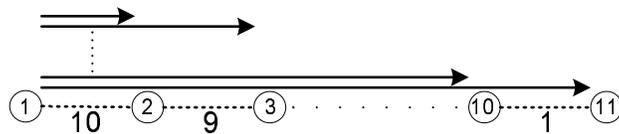
Proposition 9: FLQ-CSMA is throughput-optimal in networks where flows do not form loops.

Since the fluid limit model for the sub-system of shadow queues \hat{q} is stable from Lemma 7, the results of Propositions 8 and 9 follow the same line of analysis for the proof of Propositions 2 and 4, respectively. We omit the proofs.

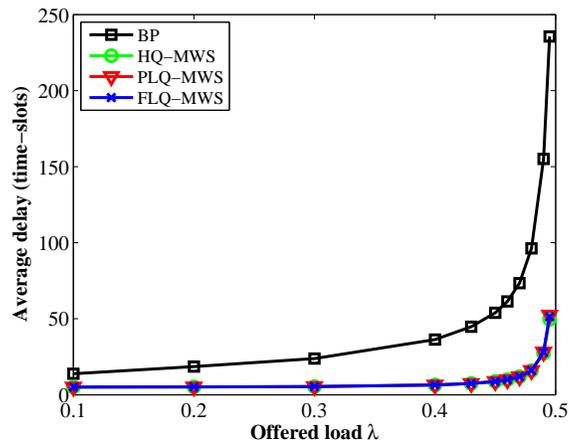
VI. NUMERICAL RESULTS

In this section, we evaluate different scheduling schemes through simulations. We compare scheduling performance of HQ-MWS, PLQ-MWS, FLQ-MWS with the original back-pressure (BP) algorithm under the *node-exclusive*⁴ interference model. Note that we focus on the node-exclusive interference model only for the purpose of illustration. Our scheduling schemes can be applied to general interference constraints as specified in Section II. We will first focus on a simple linear network topology to illustrate the advantages of the proposed schemes, and further validate our theoretical results in a larger and more

⁴It is also called the *primary* or *1-hop* interference model, where two links sharing a common node cannot be activated simultaneously. It has been known as a good representation for Bluetooth or FH-CDMA networks [2].



(a) Linear network topology with ten links



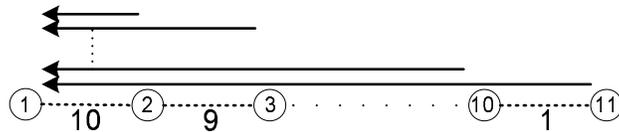
(b) Average delay

Fig. 1. Performance of BP, HQ-MWS, PLQ-MWS and FLQ-MWS in a linear network topology ($\epsilon = 0.005$).

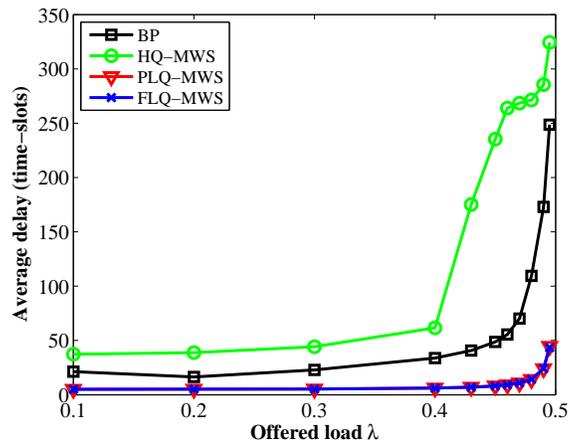
realistic grid network topology. The impact of the parameter ϵ on the scheduling performance will also be explored and discussed.

First, we evaluate and compare the scheduling performance of HQ-MWS, PLQ-MWS, FLQ-MWS and the back-pressure algorithm in a simple linear network that consists of 11 nodes and 10 links as shown in Fig. 1(a), where nodes are represented by circles and links are represented by dashed lines with link capacity, respectively. We establish 10 flows that are represented by arrows, where each flow i is from node 1 to node $i + 1$ via all the nodes in-between. We consider uniform traffic where all flows have packet arrivals at each time slot following Poisson distribution with the same mean rate $\lambda > 0$. We run our simulations with changing traffic load λ . Clearly, in this scenario, any traffic load with $\lambda < 0.5$ is feasible. We use $\epsilon = 0.005$ for HQ-MWS, PLQ-MWS and FLQ-MWS. We evaluate the performance by measuring average packet delays (in unit of time slot) over all the delivered packets (that reach their respective destination nodes) in the network.

Fig. 1(b) plots the average delays under different offered loads to examine the performance limits of different scheduling schemes. Each result represents a simulation run that lasts for 10^7 time slots. Since the optimal throughput region Λ^* is defined as the set of arrival rate vectors under which queue lengths



(a) Linear network topology with ten links

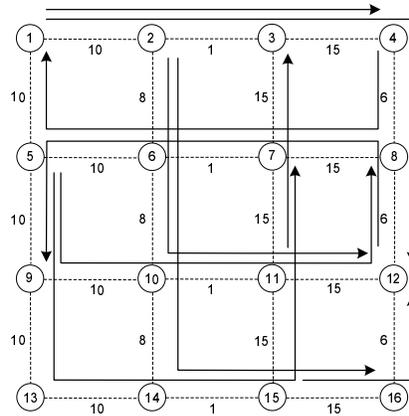


(b) Average delay

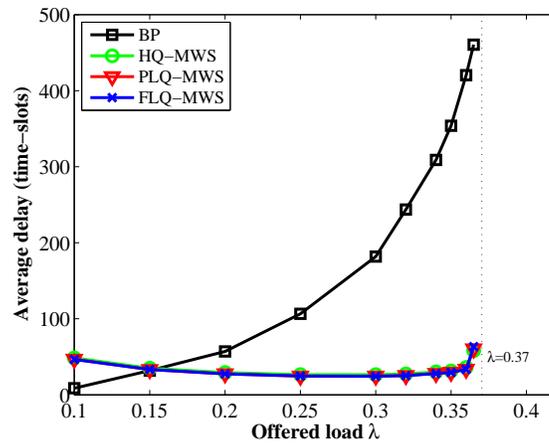
Fig. 2. Performance of BP, HQ-MWS, PLQ-MWS and FLQ-MWS in a linear network topology ($\epsilon = 0.005$).

and thus delays remain finite, we can consider the traffic load, under which the average delay increases rapidly, as the boundary of the optimal throughput region. Fig. 1(b) shows that all schemes achieve the same boundary (i.e., $\lambda < 0.5$), which supports our theoretical results on throughput optimality. Moreover, all the three proposed schemes achieve substantially better delay performance than the back-pressure algorithm. This is because under the back-pressure algorithm, the queue lengths have to build up along the route a flow takes from the destination to the source, and in general, earlier hop link has a larger queue length. This leads to poor delay performance especially when the route of a flow is lengthy, which is the case in Fig. 1(a). Note that in this specific scenario, there is only one per-hop queue at each link under HQ-MWS. Hence, HQ-MWS is equivalent to PLQ-MWS and FLQ-MWS in this scenario, which explains why the three proposed schemes perform the same as in Fig. 1(b).

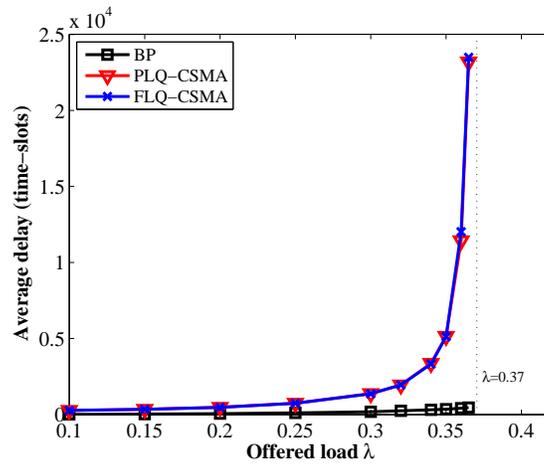
Second, we evaluate the performance of the proposed schemes in the same linear network as in the previous case while reversing the direction of each flow. The new topology is illustrated in Fig. 2(a). In this scenario, the number of per-hop queues HQ-MWS maintains for each link is the same as the number of flows passing through that link. Hence, HQ-MWS is expected to operate differently from PLQ-MWS and FLQ-MWS, and achieves different (and potentially poorer) delay performance. All the other simulation



(a) A grid network topology



(b) Average delay for MWS schemes with $\epsilon = 0.05$



(c) Average delay for CSMA schemes with $\epsilon = 0.005$

Fig. 3. Performance of all the proposed scheduling schemes in a grid network with 16 nodes and 24 links. In Fig. 3(b), the vertical dotted line $\lambda = 0.37$ denotes an upper bound for the feasible values of λ .

settings are kept the same as in the previous case. Fig. 2(b) shows that all schemes achieve the same boundary (i.e., $\lambda < 0.5$) in this scenario, which again supports our theoretical results on throughput performance. However, we observe that HQ-MWS has the worst delay performance, while PLQ-MWS and FLQ-MWS achieve substantially better performance. This is because PLQ-MWS and FLQ-MWS transmit packets more efficiently and do not waste service as long as there are enough packets at the activated link, while the back-pressure algorithm and HQ-MWS maintain multiple queues for each link, and may waste service if the activated queue has less packets than the link capacity. HQ-MWS has larger delays than the back-pressure algorithm because the scheduling decisions of HQ-MWS are based on the shadow queue lengths rather than the actual queue lengths: a queue with very small (or even zero) queue length could be activated. This introduces another type of inefficiency in HQ-MWS. Note that PLQ-MWS and FLQ-MWS also make scheduling decisions based on the shadow queue lengths. However, their performance improvement from a single queue per link dominates delay increases from the inefficiency. These observations imply that maintaining per-link queues not only simplifies the data structure, but also improves scheduling efficiency and reduces delays.

Next, we evaluate the performance of all the proposed schemes in a larger grid network with 16 nodes and 24 links as shown in Fig. 3(a), where the capacity of each link has been shown beside the link and carefully assigned to avoid traffic symmetry. Similar type of grid networks have been adopted in the literature (e.g., [4], [6], [25]) to numerically evaluate scheduling performance. We establish 10 multi-hop flows that are represented by arrows in Fig. 3(a). Again, we consider uniform traffic where each flow has independent packet arrivals at each time slot following Poisson distribution with the same mean rate $\lambda > 0$. In this scenario, we can calculate an upper bound of $1/(4/8 + 2/10 + 2) = 10/27 \approx 0.37$ for the feasible value of λ , by looking at the flows passing through node 6, which is the bottleneck in the network.

We choose $\epsilon = 0.05$ for HQ-MWS, PLQ-MWS and FLQ-MWS. Under each scheduling scheme along with the back-pressure algorithm, we measure average packet delays under different offered loads to examine their performance limits. Fig. 3(b) shows that the proposed schemes have higher packet delays than the back-pressure algorithm when traffic load is light (e.g., $\lambda < 0.15$). This is due to the aforementioned inefficiency under the proposed schemes: since the scheduling decisions are based on the shadow queue lengths rather than the actual queue lengths, queues with very small (or even zero) queue length can be activated. However, the effect tends to decrease with heavier traffic load, since the queue lengths are likely to be large. The results also show that the proposed schemes consistently outperform the back-pressure algorithm when $\lambda > 0.15$. Note that with $\epsilon = 0.05$, the shadow traffic rate

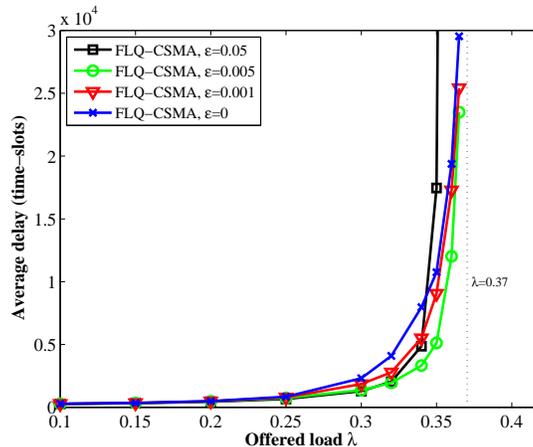


Fig. 4. The impact of the value of ϵ on the scheduling performance.

vector is outside the optimal throughput region when $\lambda > 0.37/(1 + 0.05) \approx 0.35$, however, interestingly, the schedules chosen based on the shadow queue lengths can still stabilize the data queues even if $0.35 < \lambda < 0.37$ (which is still feasible). Nevertheless, we later will show that this is not always the case. For PLQ-CSMA and FLQ-SMA, similar as in [4], we choose contention window size $W = 48$, weight function $w_l(t) = \log(0.1\hat{Q}_l(t))$, and link activation probability $p_l = \frac{e^{w_l(t)}}{e^{w_l(t)} + 1}$. We choose $\epsilon = 0.005$ for PLQ-CSMA and FLQ-CSMA, and plot their average delays over offered loads in Fig. 3(c), along with the back-pressure algorithm. Fig. 3(c) shows that although PLQ-CSMA and FLQ-CSMA achieve the optimal throughput performance, they suffer from very poor delay performance as expected. This is due to the long mixing time of the underlying Markov chain formed by the transmission schedules [4]. Note that in the above scenario, FLQ-MWS does not guarantee throughput optimality, since flows $(5 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 8)$ and $(12 \rightarrow 8 \rightarrow 7 \rightarrow 6 \rightarrow 5 \rightarrow 9)$ form a loop. However, the results in Fig. 3(b) suggest that all the schemes, including FLQ-MWS, empirically achieve the optimal throughput performance. This opens up an interesting question about throughput performance of FLQ-MWS in general settings.

Finally, we investigate sensitivity of parameter ϵ on the scheduling performance, by running simulations for PLQ-CSMA and FLQ-CSMA with different values of ϵ in the grid network in Fig. 3(a). Since the performances of PLQ-CSMA and FLQ-CSMA are very close, we report only the results for FLQ-CSMA in Fig. 4, where we plot average packet delays over the offered load λ for FLQ-CSMA with $\epsilon = 0, 0.001, 0.005$ and 0.05 , respectively. The results show that the delay performance generally improves

with a larger value of ϵ , in particular under moderate and heavy traffic loads (e.g., $\lambda > 0.25$). This is because a larger value of ϵ leads to more aggressive link activations. However, it can be observed that a larger value of ϵ (e.g., $\epsilon = 0.05$) could make the system unstable when the offered load is close to the capacity boundary (e.g., $\lambda > 0.35$). On the other hand, the impact of ϵ becomes marginal under light traffic loads (i.e., λ is small), as the inefficiency of small queue activation dominates the scheduling performance. Interestingly, although we require ϵ be positive in the analysis for throughput optimality, the simulation results show that the proposed schemes can empirically achieve the optimal throughput performance even when $\epsilon = 0$, leading to much larger delays though.

VII. CONCLUSION

In this paper, we developed scheduling policies with per-hop or per-link queues and a shadow algorithm to achieve the overall goal of removing per-flow or per-destination information requirement, simplifying queue structure, exploiting only local information, and potentially reducing delay. We showed throughput optimality of the proposed schemes that use only the readily available hop-count information, using fluid limit techniques via an inductive argument. We further simplified the solution using FIFO queueing discipline with per-link queues and showed that this is also throughput-optimal in networks without flow-loops. The problem of proving throughput optimality in general networks with algorithms (like FLQ-MWS) that use only per-link information remains an important open and challenging problem. Further, it is also worthwhile to investigate the problem with dynamic routing and see if per-flow and per-destination information can be removed even when routes are not fixed.

APPENDIX A

PROOF OF PROPOSITION 1

To begin with, let $Q(t) \triangleq [Q_{l,k}(t)]$ and $\hat{Q}(t) \triangleq [\hat{Q}_{l,k}(t)]$ denote the queue length vector and the shadow queue length vector at time slot t , respectively. We use $\|\cdot\|$ to denote the L_1 -norm of a vector, e.g., $\|Q(t)\| = \sum_{l \in \mathcal{E}} \sum_{k=1}^{L^{\max}} Q_{l,k}(t)$. We let $m_{l,k}(i)$ be the index of the flow to which the i -th packet of queue $Q_{l,k}$ belongs. In particular, $m_{l,k}(1)$ indicates the index of the flow to which the head-of-line packet of queue $Q_{l,k}$ belongs. We define the state of queue $Q_{l,k}$ at time slot t as $\mathcal{Q}_{l,k}(t) = [m_{l,k}(1), \dots, m_{l,k}(Q_{l,k}(t))]$ in an increasing order of the arriving time, or an empty sequence if $Q_{l,k}(t) = 0$. Then we denote its vector by $\mathcal{Q}(t) \triangleq [\mathcal{Q}_{l,k}(t)]$. Define $\mathbb{Z}_{\mathcal{S}} \triangleq \{1, 2, \dots, |\mathcal{S}|\}$, and let $\mathbb{Z}_{\mathcal{S}}^{\infty}$ be the set of finitely terminated sequences taking values in $\mathbb{Z}_{\mathcal{S}}$. It is evident that $\mathcal{Q}_{l,k}(t) \in \mathbb{Z}_{\mathcal{S}}^{\infty}$, and hence $\mathcal{Q}(t) \in (\mathbb{Z}_{\mathcal{S}}^{\infty})^{|\mathcal{E}| \times L^{\max}}$. We define $\mathcal{X}(t) \triangleq (\mathcal{Q}(t), \hat{Q}(t), \frac{1}{t+1}A(t))$, and then $\mathcal{X} = (\mathcal{X}(t), t \geq 0)$ is

the process describing the behavior of the underlying system. Note that in the third term of $\mathcal{X}(t)$, we use $\frac{1}{t+1}A(t)$ instead of $\frac{1}{t}A(t)$ so that it is well-defined when $t = 0$. Clearly, the evolution of \mathcal{X} forms a countable Markov chain under HQ-MWS. We abuse the notation of L_1 -norm by writing the norm of $\mathcal{X}(t)$ as $\|\mathcal{X}(t)\| \triangleq \|Q(t)\| + \lceil \|\hat{Q}(t)\| \rceil + \lceil \frac{1}{t+1}\|A(t)\| \rceil$. Let $\mathcal{X}^{(x)}$ denote a process \mathcal{X} with an initial condition such that

$$\|\mathcal{X}^{(x)}(0)\| = x. \quad (14)$$

The following Lemma was derived in [18] for continuous-time countable Markov chains, and it follows from more general results in [26] for discrete-time countable Markov chains.

Lemma 10 (Theorem 4 of [15]): Suppose that there exist a $\xi > 0$ and a finite integer $T > 0$ such that for any sequence of processes $\{\frac{1}{x}\mathcal{X}^{(x)}(xT), x = 1, 2, \dots\}$, we have

$$\limsup_{x \rightarrow \infty} \mathbb{E} \left[\frac{1}{x} \|\mathcal{X}^{(x)}(xT)\| \right] \leq 1 - \xi. \quad (15)$$

Then, the Markov chain \mathcal{X} is stable.

Lemma 10 implies the stability of the network. A stability criterion of type (15) leads to a fluid limit approach [17] to the stability problem of queueing systems. We start our analysis by establishing the *fluid limit model* as in [15], [17]. We define another process $\mathcal{Y} \triangleq (F, U, Q, \Pi, \Psi, A, D, P, \hat{Q}, \hat{\Pi}, \hat{\Psi}, \hat{A}, \hat{D}, \hat{P})$, where the tuple denotes a list of vector processes. Clearly, a sample path of $\mathcal{Y}^{(x)}$ uniquely defines the sample path of $\mathcal{X}^{(x)}$. Then we extend the definition of \mathcal{Y} to each continuous time $t \geq 0$ as $\mathcal{Y}^{(x)}(t) \triangleq \mathcal{Y}^{(x)}(\lfloor t \rfloor)$.

Recall that a sequence of functions $f_n(\cdot)$ is said to converge to a function $f(\cdot)$ *uniformly over compact (u.o.c.) intervals* if for all $t \geq 0$, $\lim_{n \rightarrow \infty} \sup_{0 \leq t' \leq t} |f_n(t') - f(t')| = 0$. Next, we consider a sequence of processes $\{\frac{1}{x_n}\mathcal{Y}^{(x_n)}(x_n \cdot)\}$ that is scaled both in time and space. Then, using the techniques of Theorem 4.1 of [17] or Lemma 1 of [15], we can show the convergence properties of the sequences in the following lemma.

Lemma 11: With probability one, for any sequence of processes $\{\frac{1}{x_n}\mathcal{Y}^{(x_n)}(x_n \cdot)\}$, where $\{x_n\}$ is a sequence of positive integers with $x_n \rightarrow \infty$, there exists a subsequence $\{x_{n_j}\}$ with $x_{n_j} \rightarrow \infty$ as $j \rightarrow \infty$

such that the following *u.o.c. convergences* hold:

$$\frac{1}{x_{n_j}} F_s^{(x_{n_j})}(x_{n_j} t) \rightarrow f_s(t), \quad (16)$$

$$\frac{1}{x_{n_j}} U_{s,k}^{(x_{n_j})}(x_{n_j} t) \rightarrow u_{s,k}(t), \quad (17)$$

$$\frac{1}{x_{n_j}} A_{l,k}^{(x_{n_j})}(x_{n_j} t) \rightarrow a_{l,k}(t), \quad (18)$$

$$\frac{1}{x_{n_j}} \hat{A}_{l,k}^{(x_{n_j})}(x_{n_j} t) \rightarrow \hat{a}_{l,k}(t), \quad (19)$$

$$\frac{1}{x_{n_j}} Q_{l,k}^{(x_{n_j})}(x_{n_j} t) \rightarrow q_{l,k}(t), \quad (20)$$

$$\frac{1}{x_{n_j}} \hat{Q}_{l,k}^{(x_{n_j})}(x_{n_j} t) \rightarrow \hat{q}_{l,k}(t), \quad (21)$$

$$\frac{1}{x_{n_j}} D_{l,k}^{(x_{n_j})}(x_{n_j} t) \rightarrow d_{l,k}(t), \quad (22)$$

$$\frac{1}{x_{n_j}} \hat{D}_{l,k}^{(x_{n_j})}(x_{n_j} t) \rightarrow \hat{d}_{l,k}(t), \quad (23)$$

$$\frac{1}{x_{n_j}} \int_0^{x_{n_j} t} \Pi_{l,k}^{(x_{n_j})}(\tau) d\tau \rightarrow \int_0^t \pi_{l,k}(\tau) d\tau, \quad (24)$$

$$\frac{1}{x_{n_j}} \int_0^{x_{n_j} t} \hat{\Pi}_{l,k}^{(x_{n_j})}(\tau) d\tau \rightarrow \int_0^t \hat{\pi}_{l,k}(\tau) d\tau, \quad (25)$$

$$\frac{1}{x_{n_j}} \int_0^{x_{n_j} t} \Psi_{l,k}^{(x_{n_j})}(\tau) d\tau \rightarrow \int_0^t \psi_{l,k}(\tau) d\tau, \quad (26)$$

$$\frac{1}{x_{n_j}} \int_0^{x_{n_j} t} \hat{\Psi}_{l,k}^{(x_{n_j})}(\tau) d\tau \rightarrow \int_0^t \hat{\psi}_{l,k}(\tau) d\tau, \quad (27)$$

$$\frac{1}{x_{n_j}} \int_0^{x_{n_j} t} P_{l,k}^{(x_{n_j})}(\tau) d\tau \rightarrow \int_0^t p_{l,k}(\tau) d\tau, \quad (28)$$

$$\frac{1}{x_{n_j}} \int_0^{x_{n_j} t} \hat{P}_{l,k}^{(x_{n_j})}(\tau) d\tau \rightarrow \int_0^t \hat{p}_{l,k}(\tau) d\tau, \quad (29)$$

where the functions $f_s, u_{s,k}, a_{l,k}, d_{l,k}, q_{l,k}, \hat{a}_{l,k}, \hat{d}_{l,k}, \hat{q}_{l,k}$ are Lipschitz continuous in $[0, \infty)$.

Note that the proof of the above lemma is quite standard using the techniques developed in [15], [17], [27]. We provide the proof in Appendix B for completeness.

Any set of limiting functions $(f, u, q, \pi, \psi, a, d, p, \hat{q}, \hat{\pi}, \hat{\psi}, \hat{a}, \hat{d}, \hat{p})$ is called a *fluid limit*. The family of these fluid limits is associated with our original stochastic network. The scaled sequences $\{\frac{1}{x_n} \mathcal{Y}^{(x_n)}(x_n \cdot)\}$ and their limits are referred to as a *fluid limit model* [16]. Since some of the limiting functions, namely $f_s, u_{s,k}, a_{l,k}, d_{l,k}, q_{l,k}, \hat{a}_{l,k}, \hat{d}_{l,k}, \hat{q}_{l,k}$, are Lipschitz continuous in $[0, \infty)$, they are absolutely continuous. Therefore, these limiting functions are differentiable at almost all time $t \in [0, \infty)$, which we call *regular time*.

Next, we will present the *fluid model equations* of the system, i.e., Eqs. (30)-(45). Fluid model equations can be thought of as belonging to a *fluid network* which is the deterministic equivalence of the original stochastic network. Any set of functions satisfying the fluid model equations is called a *fluid model*

solution of the system. We show in the following lemma that any fluid limit is a fluid model solution.

Lemma 12: Any fluid limit $(f, u, q, \pi, \psi, a, d, p, \hat{q}, \hat{\pi}, \hat{\psi}, \hat{a}, \hat{d}, \hat{p})$ satisfies the following equations:

$$f_s(t) = \lambda_s t, \quad (30)$$

$$q_{l,k}(t) = q_{l,k}(0) + a_{l,k}(t) - d_{l,k}(t), \quad (31)$$

$$a_{l,k}(t) = \sum_s H_{l,k}^s u_{s,k}(t), \quad (32)$$

$$a_{l,k}(t) = \int_0^t p_{l,k}(\tau) d\tau, \quad (33)$$

$$d_{l,k}(t) = \int_0^t \psi_{l,k}(\tau) d\tau, \quad (34)$$

$$\psi_{l,k}(t) \leq \pi_{l,k}(t), \quad (35)$$

$$\frac{d}{dt} q_{l,k}(t) = p_{l,k}(t) - \psi_{l,k}(t), \quad (36)$$

$$\frac{d}{dt} q_{l,k}(t) = \begin{cases} p_{l,k}(t) - \pi_{l,k}(t), & \text{if } q_{l,k}(t) > 0, \\ (p_{l,k}(t) - \pi_{l,k}(t))^+, & \text{otherwise,} \end{cases} \quad (37)$$

$$\hat{q}_{l,k}(t) = \hat{q}_{l,k}(0) + \hat{a}_{l,k}(t) - \hat{d}_{l,k}(t), \quad (38)$$

$$\hat{a}_{l,k}(t) = \int_0^t \hat{p}_{l,k}(\tau) d\tau, \quad (39)$$

$$\hat{d}_{l,k}(t) = \int_0^t \hat{\psi}_{l,k}(\tau) d\tau, \quad (40)$$

$$\hat{\psi}_{l,k}(t) \leq \hat{\pi}_{l,k}(t), \quad (41)$$

$$\frac{d}{dt} \hat{q}_{l,k}(t) = \hat{p}_{l,k}(t) - \hat{\psi}_{l,k}(t), \quad (42)$$

$$\frac{d}{dt} \hat{q}_{l,k}(t) = \begin{cases} \hat{p}_{l,k}(t) - \hat{\pi}_{l,k}(t), & \text{if } \hat{q}_{l,k}(t) > 0, \\ (\hat{p}_{l,k}(t) - \hat{\pi}_{l,k}(t))^+, & \text{otherwise,} \end{cases} \quad (43)$$

$$\|q(0)\| + \|\hat{q}(0)\| \leq 1, \quad (44)$$

$$\pi_{l,k}(t) = \hat{\pi}_{l,k}(t). \quad (45)$$

Proof: Note that (30) follows from the strong law of large numbers. Eqs. (31)-(35) and (38)-(41) are satisfied from the definitions. Since each of the limiting functions $q_{l,k}(t)$ is differentiable at any regular time $t \geq 0$, (36) is satisfied from (33) and (34), by taking derivative of both sides of (31). Similarly, (42) is satisfied. Further, (36) and (42) can be rewritten as (37) and (43), respectively. Eq. (44) is from the initial configuration (14), and (45) is due to the operations of HQ-MWS algorithm. ■

Due to the result of Lemma 10, we want to show that the stability criterion of (15) holds. Note that from system causality, we have $a_{l,k}(t) \leq t \sum_s H_{l,k}^s \lambda_s + \sum_s \sum_h q_{s,h}(0)$ for all link $l \in \mathcal{E}$ and all

$1 \leq k \leq L^{\max}$, for all $t \geq 0$. Then, we have

$$\begin{aligned} & \lim_{j \rightarrow \infty} \frac{1}{x_{n_j}} \|A^{(x_{n_j})}(x_{n_j} t)\| \\ & \leq \sum_l \sum_k (t \sum_s H_{l,k}^s \lambda_s + \sum_s \sum_h q_{s,h}(0)) \end{aligned}$$

almost surely, and thus,

$$\lim_{j \rightarrow \infty} \frac{1}{x_{n_j}} \left[\frac{1}{x_{n_j} t + 1} \|A^{(x_{n_j})}(x_{n_j} t)\| \right] = 0 \quad (46)$$

almost surely, for all $t \geq 0$. Therefore, it remains to be shown that the fluid limit model for the joint system of data queues and shadow queues is stable (Lemma 18). Then, by uniform integrability of the sequence $\{\frac{1}{x} \|\mathcal{X}^{(x)}(xT)\|, x = 1, 2, \dots\}$ it implies that (15) holds. We divide the proof of Lemma 18 into two parts: 1) in Lemma 15, we show that the sub-system consisting of shadow queues is stable; 2) in Lemma 17, the sub-system consisting of data queues is stable. Before proving Lemmas 15 and 17, we state and prove Lemmas 13 and 16, which are used to prove Lemmas 15 and 17, respectively.

The following lemma shows that the instantaneous shadow arrival rate is bounded in the fluid limit, and is used to show that the fluid limit model for the sub-system consisting of shadow queues is stable under HQ-MWS.

Lemma 13: For all (scaled) time $t > 0$, and for all links $l \in \mathcal{E}$ and $1 \leq k \leq L^{\max}$, with probability one, the following inequality holds,

$$\hat{\rho}_{l,k}(t) \leq (1 + \epsilon) \left(\sum_s H_{l,k}^s \lambda_s + \frac{1}{t} \right), \quad (47)$$

and in particular,

$$\hat{\rho}_{l,1}(t) = (1 + \epsilon) \sum_s H_{l,1}^s \lambda_s. \quad (48)$$

Proof: We start by stating the following lemma, which will be used to prove Lemma 13.

Lemma 14: If a sequence $\{F(n), n = 1, 2, \dots\}$ satisfies $\lim_{n \rightarrow \infty} F(n) = f$, then the following holds,

$$\lim_{n \rightarrow \infty} \frac{\sum_{\tau=1}^n F(\tau)}{n} = f.$$

Proof: We want to show that, for any $\epsilon_1 > 0$, there exists an $N < \infty$ such that $\left| \frac{\sum_{\tau=1}^n F(\tau)}{n} - f \right| < \epsilon_1$, for all $n \geq N$.

Since $\lim_{n \rightarrow \infty} F(n) = f$, then for any $\epsilon_1 > 0$, there exists a $N_1 < \infty$ such that $|F(n) - f| < \frac{\epsilon_1}{3}$, for

all $n \geq N_1$. Letting $N = \max \left\{ N_1, \frac{3(N_1-1)f}{\epsilon_1}, \frac{3 \sum_{\tau=1}^{N_1-1} F(\tau)}{\epsilon_1} \right\}$, then for all $n \geq N$, we have

$$\begin{aligned} \left| \frac{\sum_{\tau=1}^n F(\tau)}{n} - f \right| &= \left| \frac{\sum_{\tau=1}^{N_1-1} F(\tau)}{n} + \frac{\sum_{\tau=N_1}^n F(\tau)}{n} - f \right| \\ &\leq \frac{\epsilon_1}{3} + \left| \frac{\sum_{\tau=N_1}^n F(\tau)}{n} - \frac{n - N_1 + 1}{n} f \right| + \left| \frac{N_1 - 1}{n} f \right| \\ &< \frac{\epsilon_1}{3} + \frac{n - N_1 + 1}{n} \frac{\epsilon_1}{3} + \frac{\epsilon_1}{3} \leq \epsilon_1. \end{aligned} \quad (49)$$

■

Now, we prove Lemma 13. Note that we have

$$A_{l,k}(t) \leq \sum_{s \in \mathcal{S}} H_{l,k}^s F_s(t) + \sum_{i \in \mathcal{E}} \sum_{h=1}^{L^{\max}} Q_{i,h}(0), \quad (50)$$

for any $t > 0$ and for any link $l \in \mathcal{E}$ and $1 \leq k \leq L^{\max}$ due to system causality.

Since the arrival processes satisfy SLLN of type (1), we obtain from Lemma 14 that with probability one,

$$\lim_{n \rightarrow \infty} \frac{\sum_{\tau=1}^n \frac{F_s(\tau)}{\tau}}{n} = \lambda_s, \quad \text{for all } s \in \mathcal{S}. \quad (51)$$

Note that we will omit the superscript (x_{n_j}) of the random variables (depending on the choice of the sequence $\{x_{n_j}\}$) throughout the rest of the proof for notational convenience (e.g., we use $A_{l,k}(t)$ to

denote $A_{l,k}^{(x_{n_j})}(t)$. Then, for all regular time $t > 0$, all links $l \in \mathcal{E}$ and $1 \leq k \leq L^{\max}$, we have

$$\begin{aligned}
& \hat{p}_{l,k}(t) \\
&= \frac{d}{dt} \int_0^t \hat{p}_{l,k}(\tau) d\tau = \lim_{\delta \rightarrow 0} \frac{\int_0^{t+\delta} \hat{p}_{l,k}(\tau) d\tau - \int_0^t \hat{p}_{l,k}(\tau) d\tau}{\delta} \\
(29) \quad & \lim_{\delta \rightarrow 0} \lim_{j \rightarrow \infty} \frac{\sum_{\tau=\lceil tx_{n_j} \rceil}^{\lfloor (t+\delta)x_{n_j} \rfloor} \hat{P}_{l,k}(\tau)}{\delta x_{n_j}} \\
(4) \quad & \lim_{\delta \rightarrow 0} \lim_{j \rightarrow \infty} \frac{\sum_{\tau=\lceil tx_{n_j} \rceil}^{\lfloor (t+\delta)x_{n_j} \rfloor} \frac{A_{l,k}(\tau)}{\tau}}{\delta x_{n_j}} \\
(50) \quad & \leq (1 + \epsilon) \lim_{\delta \rightarrow 0} \lim_{j \rightarrow \infty} \frac{\sum_{\tau=\lceil tx_{n_j} \rceil}^{\lfloor (t+\delta)x_{n_j} \rfloor} \frac{\sum_s H_{l,k}^s F_s(\tau) + \sum_i \sum_h Q_{i,h}(0)}{\tau}}{\delta x_{n_j}} \\
&= (1 + \epsilon) \sum_s H_{l,k}^s \lim_{\delta \rightarrow 0} \lim_{j \rightarrow \infty} \frac{\sum_{\tau=1}^{\lfloor (t+\delta)x_{n_j} \rfloor} \frac{F_s(\tau)}{\tau}}{\lfloor (t + \delta)x_{n_j} \rfloor} \cdot \frac{\lfloor (t + \delta)x_{n_j} \rfloor}{\delta x_{n_j}} \\
&\quad - (1 + \epsilon) \sum_s H_{l,k}^s \lim_{\delta \rightarrow 0} \lim_{j \rightarrow \infty} \frac{\sum_{\tau=1}^{\lceil tx_{n_j} \rceil - 1} \frac{F_s(\tau)}{\tau}}{\lceil tx_{n_j} \rceil - 1} \cdot \frac{\lceil tx_{n_j} \rceil - 1}{\delta x_{n_j}} \\
&\quad + (1 + \epsilon) \lim_{\delta \rightarrow 0} \lim_{j \rightarrow \infty} \frac{\sum_i \sum_h Q_{i,h}(0)}{\delta x_{n_j}} \cdot \frac{\sum_{\tau=\lceil tx_{n_j} \rceil}^{\lfloor (t+\delta)x_{n_j} \rfloor} \frac{1}{\tau}}{\delta x_{n_j}} \\
&\leq (1 + \epsilon) \sum_s H_{l,k}^s \lambda_s \lim_{\delta \rightarrow 0} \left(\frac{t + \delta}{\delta} - \frac{t}{\delta} \right) + (1 + \epsilon) \frac{1}{t} \\
&= (1 + \epsilon) \left(\sum_s H_{l,k}^s \lambda_s + \frac{1}{t} \right),
\end{aligned}$$

where in the last inequality, the first term is from (51), and the second term is from the fact that: i) $\|q(0)\| + \|\hat{q}(0)\| \leq 1$ implies $\lim_{j \rightarrow \infty} \frac{\sum_j \sum_h Q_{j,h}(0)}{x_{n_j}} \leq 1$; and ii)

$$\begin{aligned}
& \lim_{j \rightarrow \infty} \int_{\tau=\lceil tx_{n_j} \rceil}^{\lfloor (t+\delta)x_{n_j} \rfloor} \frac{1}{\tau + 1} d\tau \leq \lim_{j \rightarrow \infty} \sum_{\tau=\lceil tx_{n_j} \rceil}^{\lfloor (t+\delta)x_{n_j} \rfloor} \frac{1}{\tau} \leq \lim_{j \rightarrow \infty} \int_{\tau=\lceil tx_{n_j} \rceil}^{\lfloor (t+\delta)x_{n_j} \rfloor} \frac{1}{\tau} d\tau \\
\iff & \lim_{j \rightarrow \infty} \log \left(\frac{\lfloor (t + \delta)x_{n_j} \rfloor + 1}{\lceil tx_{n_j} \rceil + 1} \right) \leq \lim_{j \rightarrow \infty} \sum_{\tau=\lceil tx_{n_j} \rceil}^{\lfloor (t+\delta)x_{n_j} \rfloor} \frac{1}{\tau} \leq \lim_{j \rightarrow \infty} \log \left(\frac{\lfloor (t + \delta)x_{n_j} \rfloor}{\lceil tx_{n_j} \rceil} \right) \quad (52) \\
\iff & \lim_{j \rightarrow \infty} \sum_{\tau=\lceil tx_{n_j} \rceil}^{\lfloor (t+\delta)x_{n_j} \rfloor} \frac{1}{\tau} = \log \frac{t + \delta}{t}.
\end{aligned}$$

Combining i) and ii), we have

$$\begin{aligned} & \lim_{\delta \rightarrow 0} \lim_{j \rightarrow \infty} \frac{\sum_i \sum_h Q_{i,h}(0)}{\delta x_{n_j}} \cdot \sum_{\tau=\lceil tx_{n_j} \rceil}^{\lfloor (t+\delta)x_{n_j} \rfloor} \frac{1}{\tau} \\ & \leq \lim_{\delta \rightarrow 0} \left(\frac{1}{\delta} \cdot \log \frac{t+\delta}{t} \right) = \frac{1}{t}, \end{aligned}$$

where the equality is from the L'Hospital's Rule.

So far, we have shown (47). Note that when $k = 1$, Eq. (50) reduces to $A_{l,1}(t) = \sum_{s \in \mathcal{S}} H_{l,1}^s F_s(t)$. Then, in the above derivation of $\hat{p}_{l,k}(t)$, the first inequality (which follows from (50)) becomes an equality and the right-hand side of this inequality becomes

$$(1 + \epsilon) \lim_{\delta \rightarrow 0} \lim_{j \rightarrow \infty} \frac{\sum_{\tau=\lceil tx_{n_j} \rceil}^{\lfloor (t+\delta)x_{n_j} \rfloor} \frac{\sum_s H_{l,1}^s F_s(\tau)}{\tau}}{\delta x_{n_j}}.$$

Hence, we obtain (48). ■

Remark: Lemma 13 holds when the exogenous arrival processes satisfy the SLLN, and the shadow arrivals are controlled as in (4). Note that Lemma 13 does not hold for data queues $Q_{l,k}$, since the data arrival processes do not satisfy (4) due to their dependency on the service of the previous hop queues. Lemma 13 is important to proving the stability of the shadow queues, and implies that in the fluid limit model, the instantaneous arrival rate of shadow queues is strictly inside the optimal throughput region Λ^* after a finite time.

Then, in the following lemma, we show that the fluid limit model for the sub-system consisting of shadow queues is stable⁵ under HQ-MWS.

Lemma 15: The fluid limit model for the sub-system of shadow queues \hat{q} operating under HQ-MWS satisfies that: For any $\zeta > 0$, there exists a finite $T_1 > 0$ such that for any fluid model solution with $\|\hat{q}(0)\| \leq 1$, we have that with probability one,

$$\|\hat{q}(t)\| \leq \zeta, \text{ for all } t \geq T_1,$$

for any arrival rate vector strictly inside Λ^* .

Proof: Suppose λ is strictly inside Λ^* , we can find a small $\epsilon > 0$ such that $(1 + \epsilon)\lambda$ is strictly inside Λ^* . Then, there exists a vector $\phi \in \text{Co}(\mathcal{M})$ such that $(1 + \epsilon)\lambda < \phi$, i.e., $(1 + \epsilon) \sum_s \sum_k H_{l,k}^s \lambda_s < \phi_l$, for all $l \in \mathcal{E}$. Let β denote the smallest difference between the two vectors, which is defined as $\beta \triangleq \min_{l \in \mathcal{E}} (\phi_l - (1 + \epsilon) \sum_s \sum_k H_{l,k}^s \lambda_s)$. Clearly, we have $\beta > 0$. Let T' be a finite time such that $T' >$

⁵Similar to [15], we consider a weaker criterion for the stability of the fluid limit model in Lemma 15, which can imply the stability of the original system from Lemma 10.

$\frac{(1+\epsilon)L^{\max}}{\beta}$, then we have $(1+\epsilon)\left(\sum_s \sum_k H_{l,k}^s \lambda_s + \frac{L^{\max}}{T'}\right) < \phi_l$. Let $\phi_{l,k} \triangleq (1+\epsilon)\left(\sum_s H_{l,k}^s \lambda_s + \frac{1}{T'}\right) + \frac{\phi_l - (1+\epsilon)\left(\sum_k \sum_s H_{l,k}^s \lambda_s + \frac{L^{\max}}{T'}\right)}{L^{\max}}$. Then, we have

$$\sum_k \phi_{l,k} = \phi_l, \quad (53)$$

and from (47), we have

$$\hat{p}_{l,k}(t) \leq (1+\epsilon)\left(\sum_s H_{l,k}^s \lambda_s + \frac{1}{T'}\right) < \phi_{l,k}, \quad (54)$$

for all regular time $t \geq T'$. This implies that the instantaneous arrival rate of shadow queues is strictly inside the optimal throughput region Λ^* .

We consider a quadratic-form Lyapunov function $\hat{V}(\hat{q}(t)) = \frac{1}{2} \sum_l \sum_k (\hat{q}_{l,k}(t))^2$. It is sufficient to show that for any $\zeta_1 > 0$, there exist $\zeta_2 > 0$ and a finite time $T^* > 0$ such that at any regular time $t \geq T^*$, $\hat{V}(\hat{q}(t)) \geq \zeta_1$ implies $\frac{D^+}{dt^+} \hat{V}(\hat{q}(t)) \leq -\zeta_2$. Since $\hat{q}(t)$ is differentiable for any regular time $t \geq T'$, we can obtain the derivative of $\hat{V}(\hat{q}(t))$ as

$$\begin{aligned} \frac{D^+}{dt^+} \hat{V}(\hat{q}(t)) &= \sum_l \sum_k \hat{q}_{l,k}(t) \cdot (\hat{p}_{l,k}(t) - \hat{\pi}_{l,k}(t)) \\ &= \sum_l \sum_k \hat{q}_{l,k}(t) \cdot (\hat{p}_{l,k}(t) - \phi_{l,k}) \\ &\quad + \sum_l \sum_k \hat{q}_{l,k}(t) \cdot (\phi_{l,k} - \hat{\pi}_{l,k}(t)), \end{aligned} \quad (55)$$

where $\frac{D^+}{dt^+} \hat{V}(\hat{q}(t)) = \lim_{\delta \downarrow 0} \frac{\hat{V}(\hat{q}(t+\delta)) - \hat{V}(\hat{q}(t))}{\delta}$, and the first equality is from (43).

Let us choose $\zeta_3 > 0$ such that $\hat{V}(\hat{q}(t)) \geq \zeta_1$ implies $\max_{l \in \mathcal{E}, 1 \leq k \leq L^{\max}} \hat{q}_{l,k}(t) \geq \zeta_3$. Then in the final result of (55), we can conclude that the first term is bounded. That is,

$$\begin{aligned} \sum_l \sum_k \hat{q}_{l,k}(t) \cdot (\hat{p}_{l,k}(t) - \phi_{l,k}) &\leq -\zeta_3 \min_{l,k} (\phi_{l,k} - \hat{p}_{l,k}(t)) \\ &\leq -\zeta_3 \min_{l,k} (\phi_{l,k} - (1+\epsilon)\left(\sum_s H_{l,k}^s \lambda_s + \frac{1}{T'}\right)) \triangleq -\zeta_2 < 0, \end{aligned}$$

where the second inequality is from (54). For the second term, since HQ-MWS chooses schedules that maximize the shadow queue length weighted rate, the service rate satisfies that

$$\hat{\pi}(t) \in \operatorname{argmax}_{\phi \in Co(\mathcal{M})} \sum_l \hat{q}_{l,k^*(l)}(t) \cdot \phi_l, \quad (56)$$

where i) $\hat{q}_{l,k^*(l)}(t) = \max_k \hat{q}_{l,k}(t)$, and ii) $\hat{\pi}_l(t) = \sum_k \hat{\pi}_{l,k}(t)$ with $\hat{\pi}_{l,k}(t) = 0$ when $\hat{q}_{l,k}(t) < \hat{q}_{l,k^*(l)}(t)$. This implies that $\sum_l \sum_k \hat{q}_{l,k}(t) \cdot \phi_{l,k} \leq \sum_l \sum_k \hat{q}_{l,k^*(l)}(t) \cdot \phi_{l,k} = \sum_l \hat{q}_{l,k^*(l)}(t) \cdot \phi_l \leq \sum_l \hat{q}_{l,k^*(l)}(t) \cdot \hat{\pi}_l(t) = \sum_l \sum_k \hat{q}_{l,k}(t) \cdot \hat{\pi}_{l,k}(t)$, for all $\phi \in Co(\mathcal{M})$, where the first equality and the second inequality are from (53) and (56), respectively. Then, we obtain that the second term of (55) is non-positive. This shows that $\hat{V}(\hat{q}(t)) \geq \zeta_1$ implies $\frac{D^+}{dt^+} \hat{V}(\hat{q}(t)) \leq -\zeta_2$ for all regular time $t \geq T^*$. Hence, it immediately follows that for any $\zeta > 0$, there exists a finite $T_1 \geq T^* > 0$ such that $\|\hat{q}(t)\| \leq \zeta$, for all $t \geq T_1$. ■

We next present Lemma 16 that is used to show that the sub-system consisting of data queues is stable under HQ-MWS in the fluid limit model.

Lemma 16: If data queues $q_{l,j}$ are stable for all $l \in \mathcal{E}$ and for all $j \leq k$, then there exists a finite $T_1^k > 0$ such that for all regular time $t \geq T_1^k$ and for all $l \in \mathcal{E}$, we have that with probability one,

$$\hat{p}_{l,k+1}(t) \geq (1 + \epsilon) \sum_s H_{l,k+1}^s \lambda_s.$$

The proof follows a similar argument used in the proof for Lemma 13, and is referred to Appendix C.

In the following lemma, using a hop-by-hop inductive argument, we show that the fluid model for the sub-system of data queues is stable.

Lemma 17: The fluid limit model of the sub-system of data queues q operating under HQ-MWS is stable, i.e., there exists a finite $T_2 > 0$ such that, for any fluid model solution with $\|q(0)\| \leq 1$, we have

$$\|q(t)\| = 0, \text{ for all } t \geq T_2,$$

for any arrival rate vector strictly inside Λ^* .

Proof: We prove the stability of data queues by induction.

Suppose λ is strictly inside Λ^* , the sub-system of shadow queues \hat{q} is stable from Lemma 15. Let us choose sufficiently small $\zeta > 0$ such that $\zeta < \epsilon \min_s \lambda_s$, then there exists a finite time $T_1 > 0$ such that we have $\|\hat{q}(t)\| \leq \zeta$ for any regular time $t \geq T_1$. Thus, we have $\hat{\psi}_{l,k}(t) \geq \hat{p}_{l,k}(t) - \zeta$ from (42), for all $t \geq T_1$. Hence, for all data queues and all regular time $t \geq T_1$, we have

$$\pi_{l,k}(t) = \hat{\pi}_{l,k}(t) \geq \hat{p}_{l,k}(t) - \zeta, \tag{57}$$

from (45) and (41).

Now we show by induction that all data queues are stable in the fluid limit model.

Base Case:

First, note that $\pi_{l,1}(t) \geq (1 + \epsilon) \sum_s H_{l,1}^s \lambda_s - \zeta$ from (48) and (57). Consider a sub-system that contains only queue $q_{l,1}$. From $p_{l,1}(t) = \sum_s H_{l,1}^s \lambda_s$ and (37), we have $\frac{d}{dt} q_{l,1}(t) = p_{l,1}(t) - \pi_{l,1}(t) \leq -\epsilon \sum_s H_{l,1}^s \lambda_s + \zeta < 0$, if $q_{l,1}(t) > 0$. This implies that the sub-system that contains only $q_{l,1}$ is stable, for all $l \in \mathcal{E}$.

Induction Step:

Next, we show that, if $q_{l,j}$ is stable for all $l \in \mathcal{E}$ and all $j \leq k$, then each queue $q_{l,k+1}$ is also stable for all $l \in \mathcal{E}$, where $1 \leq k < L^{\max}$.

Since $q_{l,j}(t)$ is stable for all $l \in \mathcal{E}$ and all $j \leq k$, i.e., there exists a finite $T_1^k > 0$ such that $q_{l,j}(t) = 0$ for all regular time $t \geq T_1^k$, then $u_{s,k+1}(t) = u_{s,k}(t) + q_{s,k}(0) = \dots = u_{s,1}(t) + \sum_{h \leq k} q_{s,h}(0) = \lambda_s t +$

$\sum_{h \leq k} q_{s,h}(0)$ for all $s \in \mathcal{S}$ and for all regular time $t \geq T_1^k$. Thus, we have $a_{l,k+1}(t) = t \sum_s H_{l,k+1}^s \lambda_s + \sum_s H_{l,k+1}^s \sum_{h \leq k} q_{s,h}(0)$ from (32), and $p_{l,k+1}(t) = \sum_s H_{l,k+1}^s \lambda_s$ from (33) by taking derivative, for all $l \in \mathcal{E}$ and all regular time $t \geq T_1^k$. Then, note that we have $\hat{p}_{l,k+1}(t) \geq (1 + \epsilon) \sum_s H_{l,k+1}^s \lambda_s$ from Lemma 16. Hence, we have $\pi_{l,k+1}(t) \geq (1 + \epsilon) \sum_s H_{l,k+1}^s \lambda_s - \zeta$ from (57). Therefore, we have $\frac{d}{dt} q_{l,k+1}(t) = p_{l,k+1}(t) - \pi_{l,k+1}(t) \leq -\epsilon \sum_s H_{l,k+1}^s \lambda_s + \zeta < 0$, if $q_{l,k+1}(t) > 0$. This implies that $q_{l,k+1}$ is stable for all $l \in \mathcal{E}$.

Therefore, the result follows by induction. \blacksquare

The following lemma says that the fluid limit model of joint data queues and shadow queues is stable, which follows immediately from Lemmas 15 and 17.

Lemma 18: The fluid limit model of the joint system of data queues q and shadow queues \hat{q} operating under HQ-MWS satisfies that: For any $\zeta > 0$, there exists a finite $T_2 > 0$ such that for any fluid model solution with $\|q(0)\| + \|\hat{q}(0)\| \leq 1$, we have that with probability one,

$$\|q(t)\| + \|\hat{q}(t)\| \leq \zeta, \text{ for all } t \geq T_2,$$

for any arrival rate vector strictly inside Λ^* .

Now, consider any fixed sequence of processes $\{\frac{1}{x} \mathcal{X}^{(x)}(xt), x = 1, 2, \dots\}$ (for simplicity also denoted by $\{x\}$). By Lemmas 11 and 18, we have that for any fixed $\xi_1 > 0$, we can always choose a large enough integer $T > 0$ such that for any subsequence $\{x_n\}$ of $\{x\}$, there exists a further (sub)subsequence $\{x_{n_j}\}$ such that

$$\begin{aligned} \lim_{j \rightarrow \infty} \frac{1}{x_{n_j}} (\|Q^{(x_{n_j})}(x_{n_j}T)\| + \|\hat{Q}^{(x_{n_j})}(x_{n_j}T)\|) \\ = \|q(T)\| + \|\hat{q}(T)\| \leq \xi_1 \end{aligned}$$

almost surely. This, along with (46), implies that

$$\lim_{j \rightarrow \infty} \frac{1}{x_{n_j}} \|\mathcal{X}^{(x_{n_j})}(x_{n_j}T)\| \leq \xi_1$$

almost surely, which in turn implies (for small enough ζ_1) that

$$\limsup_{x \rightarrow \infty} \frac{1}{x} \|\mathcal{X}^{(x)}(xT)\| \leq \xi_1 \triangleq 1 - \xi < 1 \quad (58)$$

almost surely. This is because there must exist a subsequence of $\{x\}$ that converges to the same limit as $\limsup_{x \rightarrow \infty} \frac{1}{x} \|\mathcal{X}^{(x)}(xT)\|$.

Next, we will show that the sequence $\{\frac{1}{x} \|\mathcal{X}^{(x)}(xT)\|, x = 1, 2, \dots\}$ is uniformly integrable. Note that link capacities are all finite (equals one, as we assumed in the system model), then for all time slots $t > 0$, we have that

$$\hat{P}_{l,k}(t) = (1 + \epsilon) \frac{A_{l,k}(t)}{t} \leq (1 + \epsilon) \frac{|\mathcal{E}|t + \sum_{s \in \mathcal{S}} F_s(t)}{t}, \quad (59)$$

for all l and k . Define a random variable

$$\Theta(T) \triangleq \frac{1}{x} \left((1 + |\mathcal{E}| \cdot L^{\max})(x + \sum_s F_s^{(x)}(xT)) + \sum_l \sum_k \sum_{\tau=1}^{xT} \hat{P}_{l,k}^{(x)}(\tau) + 2 \right).$$

Note that we have

$$\begin{aligned} \sum_l \sum_k Q_{l,k}^{(x)}(xT) &\leq x + \sum_s F_s^{(x)}(xT), \\ \hat{Q}_{l,k}^{(x)}(xT) &\leq \sum_{\tau=1}^{xT} \hat{P}_{l,k}^{(x)}(\tau) \end{aligned}$$

and

$$A_{l,k}^{(x)}(xT) \leq x + \sum_s F_s^{(x)}(xT).$$

Then, we have

$$\frac{1}{x} \|\mathcal{X}^{(x)}(xT)\| = \frac{1}{x} (\sum_l \sum_k Q_{l,k}^{(x)}(xT) + \lceil \sum_l \sum_k \hat{Q}_{l,k}^{(x)}(xT) \rceil + \lceil \frac{1}{xT+1} \sum_l \sum_k A_{l,k}^{(x)}(xT) \rceil) \leq \Theta(T),$$

and

$$\begin{aligned} \mathbb{E}[\Theta(T)] &\leq \frac{1}{x} \left((1 + |\mathcal{E}| \cdot L^{\max})(x + \sum_s \lambda_s xT) + (1 + \epsilon) \sum_l \sum_k \sum_{\tau=1}^{xT} (|\mathcal{E}| + \sum_s \lambda_s) + 2 \right) \\ &\leq \frac{1}{x} (x(1 + |\mathcal{E}| \cdot L^{\max})(1 + T \sum_s \lambda_s) + (1 + \epsilon)xT \cdot |\mathcal{E}| \cdot L^{\max}(|\mathcal{E}| + \sum_s \lambda_s) + 2) \\ &\leq (1 + |\mathcal{E}| \cdot L^{\max})(1 + T \sum_s \lambda_s) + (1 + \epsilon)T \cdot |\mathcal{E}| \cdot L^{\max}(|\mathcal{E}| + \sum_s \lambda_s) + 2 \\ &< \infty, \end{aligned}$$

where the first inequality is from (59) and the assumption on our arrival processes.

Therefore, it follows from the Dominated Convergence Theorem that the sequence $\{\frac{1}{x} \|\mathcal{X}^{(x)}(xT)\|, x = 1, 2, \dots\}$ is uniformly integrable. Then, the almost surely convergence in (58) along with uniform integrability implies the following convergence in the mean:

$$\limsup_{x \rightarrow \infty} \mathbb{E}[\frac{1}{x} \|\mathcal{X}^{(x)}(xT)\|] \leq 1 - \xi.$$

Since the above convergence holds for any sequence of processes $\{\frac{1}{x} \mathcal{X}^{(x)}(xT), x = 1, 2, \dots\}$, the condition of type (15) in Lemma 10 is satisfied. This completes the proof of Proposition 1.

APPENDIX B

PROOF OF LEMMA 11

First, we prove the convergence and continuity properties for the processes associated with data queues. It follows from the strong law of large numbers that $\frac{1}{x_n} F_s^{(x_n)}(x_n t) \rightarrow \lambda_s t$, hence, the convergence (16) holds, and each of the limiting functions f_s is Lipschitz continuous. Also, note that for any fixed

$0 \leq t_1 \leq t_2$, due to finite link capacities (in particular, all equal to one under our unit capacity assumption), we have that

$$\frac{1}{x_n} \left(D_{l,k}^{(x_n)}(x_n t_2) - D_{l,k}^{(x_n)}(x_n t_1) \right) \leq t_2 - t_1. \quad (60)$$

Thus, the sequence of functions $\{\frac{1}{x_n} D_{l,k}^{(x_n)}(x_n \cdot)\}$ is uniformly bounded and uniformly equicontinuous. Consequently, by the Arzela-Ascoli Theorem, there must exist a subsequence under which (22) holds. Note that (60) also implies that each of the limiting functions $d_{l,k}$ is Lipschitz continuous. Recall that $U_{s,k}(t)$ denotes the cumulative number of packets transmitted from the $(k-1)$ -st hop to the k -th hop for flow s up to time slot t , then convergence (17) holds similarly as (22) for $k > 1$, and holds from (16) for $k = 1$. Hence, convergence (18) trivially follows from the definition of $A_{l,k}(t)$ and (17). Similarly, each of the limiting functions $u_{s,k}$ and $a_{l,k}$ is Lipschitz continuous.

Since the sequence $\{\frac{1}{x_n} Q_{l,k}^{(x_n)}(0)\}$ are bounded by 1 from (14), there exists a further subsequence (of the subsequence already chosen above, and for simplicity still denoted by x_{n_j}) such that $\frac{1}{x_{n_j}} Q_{l,k}^{(x_{n_j})}(0) \rightarrow q_{l,k}(0)$. Hence, convergence (20) trivially follows from the queue evolution equation (3) and convergences (18) and (22). Also, it follows that each of the limiting functions $q_{l,k}$ is Lipschitz continuous.

Recall that $\Psi_{l,k}(t) = D_{l,k}(t) - D_{l,k}(t-1)$ and $P_{l,k}(t) = A_{l,k}(t) - A_{l,k}(t-1)$, hence, the sequences $\{\frac{1}{x_{n_j}} \int_0^{x_{n_j} t} \Psi_{l,k}^{(x_{n_j})}(\tau) d\tau\}$ and $\{\frac{1}{x_{n_j}} \int_0^{x_{n_j} t} P_{l,k}^{(x_{n_j})}(\tau) d\tau\}$ are identical to the sequences $\{\frac{1}{x_{n_j}} D_{l,k}^{(x_{n_j})}(x_{n_j} t)\}$ and $\{\frac{1}{x_{n_j}} A_{l,k}^{(x_{n_j})}(x_{n_j} t)\}$, respectively. This in turn implies that the convergences (26) and (22) hold, where $\int_0^t \psi_{l,k}(\tau) d\tau = d_{l,k}(t)$ and $\int_0^t p_{l,k}(\tau) d\tau = a_{l,k}(t)$. The convergence (24) follows from an inequality similar to (60) by applying the Arzela-Ascoli Theorem.

Using similar arguments, we can prove the results for the processes associated with the shadow queues. This completes the proof of Lemma 11.

APPENDIX C

PROOF OF LEMMA 16

Note that the total number of packets waiting in the previous hops for $Q_{l,k+1}$ at time slot t is no greater than $\sum_i \sum_{h \leq k} Q_{i,h}(t)$. Then, we have

$$A_{l,k+1}(t) \geq \sum_s H_{l,k+1}^s F_s(t) - \sum_i \sum_{h \leq k} Q_{i,h}(t). \quad (61)$$

Since $q_{i,h}$ is stable for all $i \in \mathcal{E}$ and all $h \leq k$, there exists a finite $T_1^k > 0$ such that $\sum_i \sum_{h \leq k} q_{i,h}(t) = 0$, for all regular time $t \geq T_1^k$. Let $\delta > 0$ be fixed, and consider all times $\nu \in [t, t + \delta]$, where $t \geq T_1^k$. Recall that x_{n_j} is a positive subsequence for which the convergence to the fluid limit holds *u.o.c.* For an

arbitrary $\theta > 0$, there exists a large enough j so that

$$\left| \frac{\sum_i \sum_{h \leq k} Q_{i,h}(x_{n_j} \nu)}{x_{n_j}} - \sum_i \sum_{h \leq k} q_{i,h}(\nu) \right| < \theta, \quad (62)$$

for all $\nu \in [t, t + \delta]$.

Consider time slots $\Upsilon \triangleq \{\lceil x_{n_j} t \rceil, \lceil x_{n_j} t \rceil + 1, \dots, \lceil x_{n_j} (t + \delta) \rceil\}$. Eq. (62) can be rewritten as

$$\sum_i \sum_{h \leq k} Q_{i,h}(\tau) < \theta x_{n_j}, \quad (63)$$

for all time slots $\tau \in \Upsilon$. Then for all $t \geq T_1^k$ and all $l \in \mathcal{E}$, we have

$$\begin{aligned} \hat{p}_{l,k+1}(t) &= \frac{d}{dt} \int_0^t \hat{p}_{l,k+1}(\tau) d\tau = \lim_{\delta \rightarrow 0} \frac{\int_0^{t+\delta} \hat{p}_{l,k+1}(\tau) d\tau - \int_0^t \hat{p}_{l,k+1}(\tau) d\tau}{\delta} \\ &= \lim_{\delta \rightarrow 0} \lim_{x_{n_j} \rightarrow \infty} \frac{\sum_{\tau=\lceil tx_{n_j} \rceil}^{\lfloor (t+\delta)x_{n_j} \rfloor} \hat{P}_{l,k+1}(\tau)}{\delta x_{n_j}} \\ &\stackrel{(a)}{=} (1 + \epsilon) \lim_{\delta \rightarrow 0} \lim_{x_{n_j} \rightarrow \infty} \frac{\sum_{\tau=\lceil tx_{n_j} \rceil}^{\lfloor (t+\delta)x_{n_j} \rfloor} \frac{A_{l,k+1}(\tau)}{\tau}}{\delta x_{n_j}} \\ &\stackrel{(b)}{\geq} (1 + \epsilon) \lim_{\delta \rightarrow 0} \lim_{x_{n_j} \rightarrow \infty} \frac{\sum_{\tau=\lceil tx_{n_j} \rceil}^{\lfloor (t+\delta)x_{n_j} \rfloor} \frac{\sum_s H_{l,k+1}^s F_s(\tau) - \sum_i \sum_{h \leq k} Q_{i,h}(\tau)}{\tau}}{\delta x_{n_j}} \\ &\stackrel{(c)}{>} (1 + \epsilon) \sum_s H_{l,k+1}^s \lim_{\delta \rightarrow 0} \lim_{x_{n_j} \rightarrow \infty} \frac{\sum_{\tau=1}^{\lfloor (t+\delta)x_{n_j} \rfloor} \frac{F_s(\tau)}{\tau}}{\lfloor (t + \delta)x_{n_j} \rfloor} \cdot \frac{\lfloor (t + \delta)x_{n_j} \rfloor}{\delta x_{n_j}} \\ &\quad - (1 + \epsilon) \sum_s H_{l,k+1}^s \lim_{\delta \rightarrow 0} \lim_{x_{n_j} \rightarrow \infty} \frac{\sum_{\tau=1}^{\lceil tx_{n_j} \rceil - 1} \frac{F_s(\tau)}{\tau}}{\lceil tx_{n_j} \rceil - 1} \cdot \frac{\lceil tx_{n_j} \rceil - 1}{\delta x_{n_j}} \\ &\quad - (1 + \epsilon) \lim_{\delta \rightarrow 0} \lim_{x_{n_j} \rightarrow \infty} \left(\frac{1}{\delta} \cdot \frac{\theta x_{n_j}}{x_{n_j}} \cdot \sum_{\tau=\lceil tx_{n_j} \rceil}^{\lfloor (t+\delta)x_{n_j} \rfloor} \frac{1}{\tau} \right) \\ &\stackrel{(d)}{=} (1 + \epsilon) \sum_s H_{l,k+1}^s \lambda_s \lim_{\delta \rightarrow 0} \left(\frac{t + \delta}{\delta} - \frac{t}{\delta} \right) - (1 + \epsilon) \lim_{\delta \rightarrow 0} \left(\frac{\theta}{\delta} \cdot \log \frac{t + \delta}{t} \right) \\ &= (1 + \epsilon) \left(\sum_s H_{l,k+1}^s \lambda_s - \frac{\theta}{t} \right), \end{aligned}$$

where (a), (b) and (c) are from (4), (61) and (63), respectively, and (d) is from (51) and (52).

Since $\theta > 0$ can be arbitrary, we complete the proof by letting $\theta \rightarrow 0$.

APPENDIX D

STABILITY OF THE SHADOW QUEUES UNDER LQ-MWS

Similarly to (16)-(29), we can establish the fluid limits of the system: $(f, u, q, \pi, \psi, a, d, p, \hat{q}, \hat{\pi}, \hat{\psi}, \hat{a}, \hat{d}, \hat{p})$, and we have the following fluid model equations:

$$f_s(t) = \lambda_s t, \quad (64)$$

$$q_l(t) = q_l(0) + a_l(t) - d_l(t), \quad (65)$$

$$a_l(t) = \sum_s \sum_k H_{l,k}^s u_{s,k}(t), \quad (66)$$

$$a_l(t) = \int_0^t p_l(\tau) d\tau, \quad (67)$$

$$d_l(t) = \int_0^t \psi_l(\tau) d\tau, \quad (68)$$

$$\psi_l(t) \leq \pi_l(t), \quad (69)$$

$$\frac{d}{dt} q_l(t) = p_l(t) - \psi_l(t), \quad (70)$$

$$\frac{d}{dt} q_l(t) = \begin{cases} p_l(t) - \pi_l(t), & \text{if } q_l(t) > 0, \\ (p_l(t) - \pi_l(t))^+, & \text{otherwise,} \end{cases} \quad (71)$$

$$\hat{q}_l(t) = \hat{q}_l(0) + \hat{a}_l(t) - \hat{d}_l(t), \quad (72)$$

$$\hat{a}_l(t) = \int_0^t \hat{p}_l(\tau) d\tau, \quad (73)$$

$$\hat{d}_l(t) = \int_0^t \hat{\psi}_l(\tau) d\tau, \quad (74)$$

$$\hat{\psi}_l(t) \leq \hat{\pi}_l(t), \quad (75)$$

$$\frac{d}{dt} \hat{q}_l(t) = \hat{p}_l(t) - \hat{\psi}_l(t), \quad (76)$$

$$\frac{d}{dt} \hat{q}_l(t) = \begin{cases} \hat{p}_l(t) - \hat{\pi}_l(t), & \text{if } \hat{q}_l(t) > 0, \\ (\hat{p}_l(t) - \hat{\pi}_l(t))^+, & \text{otherwise,} \end{cases} \quad (77)$$

$$\|q(0)\| + \|\hat{q}(0)\| = 1, \quad (78)$$

$$\pi_l(t) = \hat{\pi}_l(t). \quad (79)$$

We present a lemma similar to Lemma 13. This will be used to show that the fluid limit model for the sub-system consisting of shadow queues is stable under LQ-MWS. We omit its proof since it follows the same line of analysis for the proof of Lemma 13.

Lemma 19: For all (scaled) time $t > 0$ and for all links $l \in \mathcal{E}$, we have that with probability one,

$$\hat{p}_l(t) \leq (1 + \epsilon) \left(\sum_s \sum_k H_{l,k}^s \lambda_s + \frac{1}{t} \right). \quad (80)$$

Now, we can show that the fluid limit model for the sub-system of shadow queues \hat{q} is stable under LQ-MWS.

Lemma 20: The fluid limit model for the sub-system of shadow queues \hat{q} operating under LQ-MWS satisfies that: For any $\zeta > 0$, there exists a finite $T_3 > 0$ such that for any fluid model solution with $\|\hat{q}(0)\| \leq 1$, we have that with probability one,

$$\|\hat{q}(t)\| \leq \zeta, \text{ for all } t \geq T_3, \quad (81)$$

for any arrival rate vector strictly inside Λ^* .

The proof is similar to that of Lemma 15 and is thus omitted.

APPENDIX E

PROOF OF PROPOSITION 2

To show the stability of the network under PLQ-MWS, it is enough to show that the fluid limit model of the joint system of data queues and shadow queues is stable. Since the fluid limit model for the sub-system of shadow queues is stable from Lemma 20, it remains to show that the fluid model for the sub-system of data queues is stable, i.e., it is equivalent to show that all the sub-queues for hop-class k packets are stable for each $1 \leq k \leq L^{\max}$. We will prove the stability of sub-queues via a hop-by-hop inductive argument.

Let $Q_{l,k}(t)$ denote the number of packets of hop-class k at Q_l at time slot t , and let $A_{l,k}(t)$, $D_{l,k}(t)$, $\Pi_{l,k}(t)$, $\Psi_{l,k}(t)$ and $P_{l,k}(t)$ denote the cumulative arrival, cumulative departure, service, departure and arrival for packets of hop-class k at Q_l , respectively. As before, we establish the fluid limits of the system, and obtain (64)-(79) and the following additional fluid model equations: for all (scaled) time $t \geq 0$,

$$a_{l,k}(t) = \sum_s H_{l,k}^s u_{s,k}(t), \quad (82)$$

$$a_{l,k}(t) = \int_0^t p_{l,k}(\tau) d\tau, \quad (83)$$

$$\frac{d}{dt} q_{l,k}(t) = p_{l,k}(t) - \psi_{l,k}(t), \quad (84)$$

$$\frac{d}{dt} q_{l,k}(t) = \begin{cases} p_{l,k}(t) - \pi_{l,k}(t), & \text{if } q_{l,k}(t) > 0, \\ (p_{l,k}(t) - \pi_{l,k}(t))^+, & \text{otherwise.} \end{cases} \quad (85)$$

Clearly, packets of hop-class k at link l will not be transmitted under PLQ-MWS unless link l is active at time slot t and $\sum_{j < k} Q_{l,j}(t) < c_l$ (Equivalently, $Q_{l,j}(t) = 0$ for all $j < k$ in our setting, since $c_l = 1$), i.e., for all $1 \leq k \leq L^{\max}$, we have

$$\Pi_{l,k}(t) = \left(\Pi_l(t) - \sum_{j < k} Q_{l,j}(t) \right)^+, \quad (86)$$

where $\Pi_l(t) = 1$, if link l is active at time slot t , and $\Pi_l(t) = 0$, otherwise. Hence, we have an additional fluid model equation as follows:

$$\pi_{l,k}(t) = \pi_l(t) - \sum_{j < k} \psi_{l,j}(t), \quad (87)$$

for all $1 \leq k \leq L^{\max}$, and in particular, we have

$$\pi_{l,1}(t) = \pi_l(t), \quad (88)$$

for all $l \in \mathcal{E}$ and for all $t \geq 0$.

From Lemma 20, the fluid limit model for the sub-system consisting of shadow queues is stable, i.e., there exists a finite $T_3 > 0$ such that, for all $l \in \mathcal{E}$ and for all time $t \geq T_3$,

$$\pi_l(t) = \hat{\pi}_l(t) \geq \hat{p}_l(t). \quad (89)$$

Next, we show the stability of sub-queues by induction.

Base Case:

We first show that sub-queues $q_{l,1}$ are stable for all $l \in \mathcal{E}$. Note that $\mathbb{E}[\hat{P}_l(t)] = (1 + \epsilon) \frac{A_l(t)}{t} \geq (1 + \epsilon) \frac{\sum_s H_{l,1}^s F_s(t)}{t}$, and following the same line of analysis for the proof of Lemma 16, we show that,

$$\hat{p}_l(t) \geq (1 + \epsilon) \sum_s H_{l,1}^s \lambda_s,$$

for all $t \geq 0$. This, along with (88) and (89), implies that

$$\pi_{l,1}(t) \geq (1 + \epsilon) \sum_s H_{l,1}^s \lambda_s,$$

for all $l \in \mathcal{E}$ and for all time $t \geq T_3$.

Consider the sub-system that only contains sub-queue $q_{l,1}$, and note that $p_{l,1}(t) = \sum_s H_{l,1}^s \lambda_s$, then for all $t \geq T_3$, we have $\frac{d}{dt} q_{l,1}(t) = p_{l,1}(t) - \pi_{l,1}(t) \leq -\epsilon \sum_s H_{l,1}^s \lambda_s < 0$, if $q_{l,1}(t) > 0$. This implies that the sub-system that consists of $q_{l,1}$ is stable, for all $l \in \mathcal{E}$.

Induction Step:

Next, we show that, if sub-queues $q_{l,j}$ for all $l \in \mathcal{E}$ and all $j \leq k$ is stable, then each sub-queue $q_{l,k+1}$ for all $l \in \mathcal{E}$ is also stable, along with the stability of $q_{l,j}$ for all $l \in \mathcal{E}$ and all $j \leq k$.

Recall that $U_{s,k}(t)$ is the number of packets transmitted from the $(k-1)$ -st hop to the k -th hop for flow s up to time slot t , and $u_{s,k}(t)$ is its fluid limit. Since $q_{l,j}(t)$ is stable for all $l \in \mathcal{E}$ and all $j \leq k$, i.e., there exists a finite $T_2^k > 0$ such that $q_{l,j}(t) = 0$ for all regular time $t \geq T_2^k$, then $u_{s,k+1}(t) = u_{s,k}(t) + q_{s,k}(0) = \dots = u_{s,1}(t) + \sum_{h \leq k} q_{s,h}(0) = \lambda_s t + \sum_{h \leq k} q_{s,h}(0)$ for all $s \in \mathcal{S}$, for all regular time $t \geq T_2^k$. Thus, for all $l \in \mathcal{E}$ and for all $j \leq k+1$, we have $a_{l,j}(t) = t \sum_s H_{l,j}^s \lambda_s + \sum_s H_{l,j}^s \sum_{h < j} q_{s,h}(0)$

from (82), and $p_{l,j}(t) = \sum_s H_{l,j}^s \lambda_s$ from (83) by taking derivative, for all $l \in \mathcal{E}$ and all regular time $t \geq T_1^k$. Hence, from (84) and the stability of $q_{l,j}$ (i.e., $\frac{d}{dt}q_{l,j}(t) = 0$) for all $j \leq k$, we have that for all $j \leq k$,

$$\psi_{l,j}(t) = p_{l,j}(t) = \sum_s H_{l,j}^s \lambda_s. \quad (90)$$

Note that since

$$\mathbb{E}[\hat{P}_l(t)] = (1 + \epsilon) \frac{A_l(t)}{t} \geq (1 + \epsilon) \frac{\sum_s \sum_{j \leq k+1} H_{l,j}^s A_{l,j}(t)}{t},$$

we can obtain that

$$\hat{p}_l(t) \geq (1 + \epsilon) \sum_s \sum_{j \leq k+1} H_{l,j}^s \lambda_s, \quad (91)$$

following the same line of analysis of Lemma 16. Hence, from (87), (89), (90) and (91), we have that for all $j \leq k$,

$$\pi_{l,k+1}(t) \geq (1 + \epsilon) \sum_s H_{s,k+1}^l \lambda_s + \epsilon \sum_s \sum_{j \leq k} H_{l,j}^s \lambda_s. \quad (92)$$

This implies that for all time $t \geq T_2^k$, $\frac{d}{dt}q_{l,k+1}(t) = p_{l,k+1}(t) - \pi_{l,k+1}(t) \leq -\epsilon \sum_s \sum_{j \leq k+1} H_{l,j}^s \lambda_s < 0$, if $q_{l,k+1}(t) > 0$. Hence, we can conclude that $q_{l,k+1}$ is stable for all $l \in \mathcal{E}$.

Now by induction, we can show that all the data queues in fluid limits are stable. With Lemma 20, this implies that the fluid limit model of the joint system of data queues and shadow queues is stable. Then, we can conclude Proposition 2 following the same arguments used in the proof of Proposition 1.

APPENDIX F

PROOF OF LEMMA 3

Recall that $\mathcal{L}(s)$ denotes the loop-free route of the flow s . We prove Lemma 3 in a constructive way, i.e., for a network where flows do not form loops, we will give an algorithm that generates a ranking such that the following statements in Lemma 3 hold: 1) for any flow $s \in \mathcal{S}$, the ranks are monotonically increasing when one traverses the links on the route of the flow s from l_1^s to $l_{|\mathcal{L}(s)|}^s$, i.e., $r(l_i^s) < r(l_{i+1}^s)$ for all $1 \leq i < |\mathcal{L}(s)|$; and 2) the packet arrivals at a link are either exogenous, or forwarded from links with a smaller rank.

We start with some useful definitions.

Definition 1: Two flows $s_1, s_2 \in \mathcal{S}$ are *connected*, if they have common (directed) links on their routes, i.e., $\mathcal{L}(s_1) \cap \mathcal{L}(s_2) \neq \emptyset$, and *disconnected*, otherwise. A sequence of flows (τ_1, \dots, τ_n) is a *communicating sequence*, if every two adjacent flows τ_i and τ_{i+1} are connected with each other. Two flows s_1 and s_2 *communicate*, if there exists a communicating sequence between s_1 and s_2 .

Definition 2: Let $\mathcal{S}(l) \subseteq \mathcal{S}$ denote the set of flows passing through link l , and let $\mathcal{S}(\mathcal{Z}) \triangleq \bigcup_{l \in \mathcal{Z}} \mathcal{S}(l)$ denote the set of flows passing through a set of links $\mathcal{Z} \subseteq \mathcal{E}$. A non-empty set of links \mathcal{Z} is called a *component*, if the following conditions are satisfied:

- 1) $\mathcal{Z} = \bigcup_{s \in \mathcal{S}(\mathcal{Z})} \mathcal{L}(s)$.
- 2) Either $|\mathcal{S}(\mathcal{Z})| = 1$, or any two flows $s_1, s_2 \in \mathcal{S}(\mathcal{Z})$ communicate.

Definition 3: Consider a component \mathcal{Z} , a sequence⁶ of flows $(s_1, s_2, \dots, s_N) \subseteq \mathcal{S}(\mathcal{Z})$, where $N \geq 2$, is said to form a *flow-loop*, if one can find two links $l_{i_n}^{s_n}$ and $l_{j_n}^{s_n}$ for each $n = 1, 2, \dots, N$, satisfying

- 1) $i_n < j_n$ for each $1 \leq n \leq N$,
- 2) $\begin{cases} l_{j_n}^{s_n} = l_{i_{n+1}}^{s_{n+1}} \text{ for each } n < N, \\ l_{j_N}^{s_N} = l_{i_1}^{s_1}. \end{cases}$

An example of a component that contains a flow-loop is presented in Fig. 5(a), where the network consists of seven links and six flows. The routes of the flows are as follows: $\mathcal{L}(s_1) = (1, 2, 3)$, $\mathcal{L}(s_2) = (3, 4)$, $\mathcal{L}(s_3) = (4, 5)$, $\mathcal{L}(s_4) = (5, 6)$, $\mathcal{L}(s_5) = (6, 7)$, $\mathcal{L}(s_6) = (7, 2)$.

Definition 4: A component \mathcal{Z} is called a *flow-tree*, if \mathcal{Z} does not contain any flow-loops.

Definition 5: Consider a component \mathcal{Z} , a link $l \in \mathcal{Z}$ is called a *starting link*, if there exists a flow $s' \in \mathcal{S}(\mathcal{Z})$ such that $H_{l,1}^{s'} = 1$ and $H_{l,k}^s = 0$ for all other $s \in \mathcal{S}(\mathcal{Z})$ and all $k \geq 2$, i.e., a starting link has only exogenous arrivals. Similarly, a link $l \in \mathcal{Z}$ is called an *ending link*, if there exists a flow $s'' \in \mathcal{S}(\mathcal{Z})$ such that, $H_{l,|\mathcal{L}(s'')|}^{s''} = 1$, and $H_{l,k}^s = 0$ for all other $s \in \mathcal{S}(\mathcal{Z})$ and all $k < |\mathcal{L}(s)|$, i.e., an ending link transmits only packets that will leave the system immediately. A path $P = (l_{P,1}, l_{P,2}, \dots, l_{P, \text{len}(P)})$, where $\text{len}(P)$ denotes the length of path P and $l_{P,i}$ denotes the i -th hop link of P , is called a *flow-path*, if the following conditions are satisfied:

- 1) Links $l_{P,1}$ and $l_{P, \text{len}(P)}$ are the only starting and ending link on the path P , respectively.
- 2) Either $\text{len}(P) = 1$, or for each $1 \leq i < \text{len}(P)$, there exists a flow s such that, $l_{P,i} \in \mathcal{L}(s)$ and $l_{P,i+1} \in \mathcal{L}(s)$, i.e., two adjacent links $l_{P,i}$ and $l_{P,i+1}$ are on the route of some flow.

In general, a flow-tree consists of multiple (possibly overlapped) flow-paths. An illustration of flow-loop, flow-path, and flow-tree is presented in Fig. 5. It is clear from Definition 3 that, if there exists a flow-loop in a component, this component must contain a cycle of links, while the opposite is not necessarily true. For example, the components in Figs. 5(b) and 5(c) both contain a cycle, while neither of them contains a flow-loop.

⁶By slightly abusing the notation, we also use (s_1, s_2, \dots, s_N) to denote the set of unique elements of the sequence.

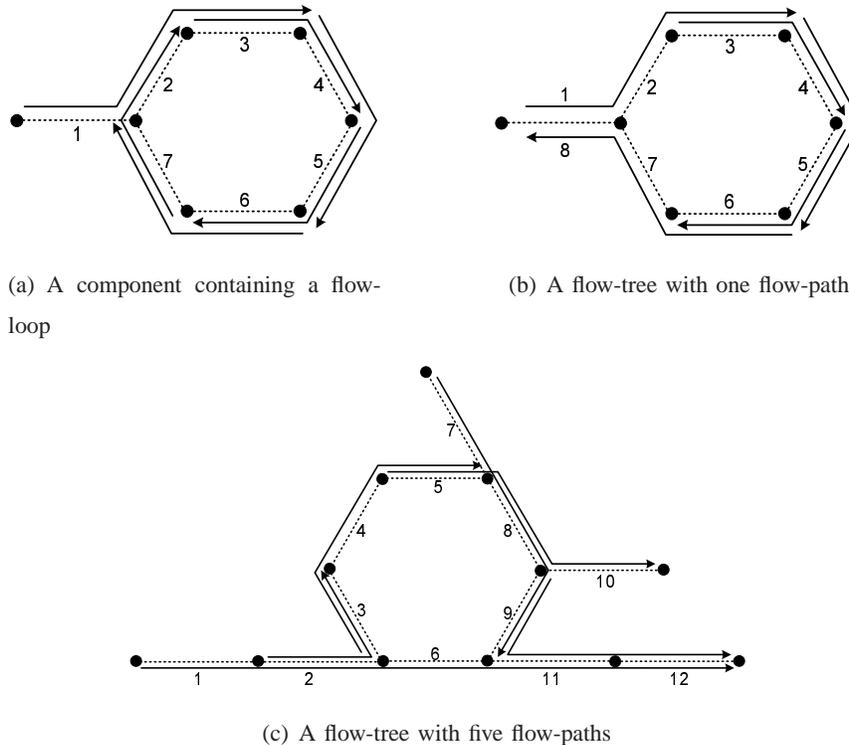


Fig. 5. Examples of different types of components. Links and flows are denoted by dashed lines with numbers and solid lines with arrows, respectively. Note that links without data flows are omitted (not numbered), and two numbers labeled beside a dashed line stand for two links with opposite directions, e.g., links 1 and 8 in Fig. 5(b). In Fig. 5(a), all flows together forms a flow-loop (2, 3, 4, 5, 6, 7), and the component is not a flow-tree. In Fig. 5(b), the component is a flow-tree and consists of one single flow-path: (1, 2, 3, 4, 5, 6, 7, 8). In Fig. 5(c), the component is a flow-tree and consists of five flow-paths: $P_1 = (1, 2, 3, 4, 5, 8, 10)$, $P_2 = (1, 2, 6, 11, 12)$, $P_3 = (7, 8, 10)$, $P_4 = (7, 8, 9, 11, 12)$ and $P_5 = (1, 2, 3, 4, 5, 8, 9, 11, 12)$.

Now, we describe Algorithm 2, which is used to generate a ranking for a network without flow-loops such that the monotone property in Lemma 3 holds.

Let $\mathcal{E}(P)$ denote the set of links belonging to flow-path P . Let \mathcal{T} denote a flow-tree, and let $\mathcal{P}(\mathcal{T})$ denote the set of all flow-paths in \mathcal{T} , i.e., $\mathcal{P}(\mathcal{T}) \triangleq \{P \text{ is a flow-path} \mid \mathcal{E}(P) \subseteq \mathcal{T}\}$. Let $P_k(\mathcal{T})$ denote the flow-path chosen in the k -th while-loop when running Algorithm 2 for \mathcal{T} , and let $\mathcal{P}_k(\mathcal{T}) \triangleq \bigcup_{j < k} P_j(\mathcal{T})$. Let $r(l)$ denote the rank of link $l \in \mathcal{T}$, and let $\mathcal{P}(l)$ denote the set of flow-paths passing through link l , i.e., $\mathcal{P}(l) \triangleq \{P \in \mathcal{P}(\mathcal{T}) \mid l \in \mathcal{E}(P)\}$. Let $\Gamma_k(l) \triangleq \{l' \in \bigcup_{P \in \mathcal{P}(l) \cap \mathcal{P}_k(\mathcal{T})} \mathcal{E}(P) \mid r(l') > r(l)\}$ denote the set of links that belong to the flow-paths of $\mathcal{P}(l) \cap \mathcal{P}_k(\mathcal{T})$ (i.e., flow-paths that pass through link l and are chosen in the j -th while-loop for $j < k$) and have a rank greater than $r(l)$.

The details of ranking are provided in Algorithm 1. In line 2, we do initialization by setting the rank of all links of \mathcal{T} to -1 . In lines 4-21, we pick a flow-path $P \in \mathcal{P}'$, and assign a rank to each link of

Algorithm 2 Rank Assignment

```

1: procedure ASSIGNRANK( $\mathcal{T}$ )
2:    $r(l) \leftarrow -1$  for all  $l \in \mathcal{T}$ 
3:    $\mathcal{P}' \leftarrow \mathcal{P}(\mathcal{T})$ 
4:   while  $\mathcal{P}' \neq \emptyset$  do
5:     pick a flow-path  $P \in \mathcal{P}'$ 
6:      $count \leftarrow 1$ 
7:     for  $1 \leq i \leq \text{len}(P)$  do
8:       if  $r(l_{P,i}) = -1$  then
9:          $r(l_{P,i}) \leftarrow count$ 
10:      else if  $r(l_{P,i}) \geq count$  then
11:         $count \leftarrow r(l_{P,i})$ 
12:      else
13:        for all  $l \in \Gamma_k(l_{P,i})$  do
14:           $r(l) \leftarrow r(l) + (count - r(l_{P,i}))$ 
15:        end for
16:         $r(l_{P,i}) \leftarrow count$ 
17:      end if
18:       $count \leftarrow count + 1$ 
19:    end for
20:     $\mathcal{P}' \leftarrow \mathcal{P}' \setminus \{P\}$ 
21:  end while
22: end procedure

```

P starting from link $l_{P,1}$. We may update a link's rank if we already assigned a rank to that link. The set of flow-paths \mathcal{P}' is updated in line 20. The while-loop continues until \mathcal{P}' becomes empty. We set $count = 1$ in line 6, and assign a rank to links $l_{P,i}$ for each $1 \leq i \leq \text{len}(P)$. For each link $l_{P,i}$, we consider the following three cases: 1) $r(l_{P,i}) = -1$; 2) $r(l_{P,i}) \geq count$; 3) $0 < r(l_{P,i}) < count$.

Case 1): link $l_{P,i}$ has not been assigned a rank yet. We set $r(l_{P,i}) = count$ in line 9.

Case 2): link $l_{P,i}$ already has a rank that is no smaller than the current $count$. In this case, the rank does not need an update, and we set $count = r(l_{P,i})$ in line 11.

TABLE I
THE EVOLUTION OF THE RANKING FOR THE FLOW-TREE IN FIG. 5(C)

Iteration k	Ranking of links 1 – 12
0	(-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1)
1	(1,2,3,4,5,-1,-1,6,-1,7,-1,-1)
2	(1,2,3,4,5,3,-1,6,-1,7,4,5)
3	(1,2,3,4,5,3,1,6,-1,7,4,5)
4	(1,2,3,4,5,3,1,6,7,7,8,9)
5	(1,2,3,4,5,3,1,6,7,7,8,9)

Case 3): link $l_{P,i}$ already has a rank that is smaller than the current *count*. In this case, we update the rank of some other links as well as that of link $l_{P,i}$. Specifically, for all the links $l \in \Gamma_k(l_{P,i})$, i.e., links that belong to the flow-paths in $\mathcal{P}(l) \cap \mathcal{P}_k(\mathcal{T})$ and have a rank greater than $r(l_{P,i})$, we increase their ranks by $\text{count} - r(l_{P,i})$ in lines 13-15. Then, we update the rank of link $l_{P,i}$ by setting it to *count* in line 16.

After considering all three cases, we increase the value of *count* by 1 in line 18.

The intention of this ranking is to assign a rank to each link such that the ranks are monotonically increasing when one traverses any flow-path from its starting link. Algorithm 2 may give different ranking to a given flow-tree depending on the order of choosing flow-paths. We give two examples for illustration as follows. In Fig. 5(b), one (and the unique one in this case) example of the ranking for the flow-tree is (1, 2, 3, 4, 5, 6, 7, 8) for links 1-8. In Fig. 5(c), one example of the ranking for the flow-tree is (1, 2, 3, 4, 5, 3, 1, 6, 7, 7, 8, 9) for links 1-12. The evolution of the ranking for the flow-tree in Fig. 5(c) is presented in Table I, where flow-path P_i is chosen in the i -th while-loop, for $i = 1, 2, 3, 4, 5$.

Since we assume $\sum_s \sum_{k=1}^{|\mathcal{L}(s)|} H_{l,k}^s \geq 1$ for all $l \in \mathcal{E}$, a network graph \mathcal{G} can be decomposed into multiple disjoint components. Clearly, a network with no flow-loops is equivalent to that all the components of the network are flow-trees. Without loss of generality, in the rest of the proof, we assume that the network that we consider consists of one single component, which is a flow-tree under the condition of Lemma 3. The same argument applies to the case with multiple disjoint components. We claim the following lemma and provide its proof in Appendix G.

Lemma 21: Algorithm 2 assigns a rank to each link of a flow-tree \mathcal{T} such that for any flow-path $P \in \mathcal{P}(\mathcal{T})$, the ranks are monotonically increasing when one traverses the links of P from $l_{P,1}$ to $l_{P,\text{len}(P)}$, i.e., $r(l_{P,i}) < r(l_{P,i+1})$ for all $1 \leq i < \text{len}(P)$ and for any $P \in \mathcal{P}(\mathcal{T})$.

Now, consider any flow $s \in \mathcal{S}$. The statement 1) holds trivially for the case of $|\mathcal{L}(s)| = 1$. Hence, we assume that $|\mathcal{L}(s)| > 1$. It is clear that for any $1 \leq i < |\mathcal{L}(s)|$, the links l_i^s and l_{i+1}^s must belong to some flow-path $P \in \mathcal{P}(\mathcal{E})$, where \mathcal{E} is assumed to be a flow-tree. Therefore, the statement 1) follows from Lemma 21.

Note that the packet arrivals at a link are either exogenous or from the previous hop on the route of some flow passing through it. Owing to the monotonically increasing rank assignment, it is clear that these previous hop links have a smaller rank. Hence, the statement 2) immediately follows from statement 1). This completes the proof of Lemma 3.

APPENDIX G

PROOF OF LEMMA 21

We want to show that Algorithm 2 assigns a rank to each link of flow-tree \mathcal{T} satisfying that $r(l_{P,i}) < r(l_{P,i+1})$, for all $1 \leq i < \text{len}(P)$ and for any $P \in \mathcal{P}(\mathcal{T})$. We use the method of induction.

Recall that $P_k(\mathcal{T})$ denotes the flow-path chosen in the k -th while-loop, and $\mathcal{P}_k(\mathcal{T}) = \bigcup_{j < k} P_j(\mathcal{T})$. We denote $P_k(\mathcal{T})$ and $\mathcal{P}_k(\mathcal{T})$ by P_k and \mathcal{P}_k , respectively, whenever there is no confusion.

Base Case:

It is trivial for the case of $k = 1$. Since we initialize $r(l_{P_1,i}) = -1$ for all $1 \leq i \leq \text{len}(P_1)$, we should have $r(l_{P_1,i}) = i$ for all $1 \leq i \leq \text{len}(P_1)$ from lines 9 and 18 of Algorithm 2, after running the first while-loop.

Induction Step:

We show that after running the k -th while-loop of Algorithm 2, if

$$r(l_{P_j,i}) < r(l_{P_j,i+1}) \text{ for all } 1 \leq i < \text{len}(P_j) \text{ and for all } j \leq k, \quad (93)$$

then after running the $(k + 1)$ -st while-loop the same result holds for all $j \leq k + 1$. In other words, once Algorithm 2 assigns the ranks for links of a flow-path in a monotonically increasing way, then this property does not change afterward. We also prove this induction step using method of induction.

We first show that if (93) holds, then after the first iteration (for assigning a rank to link $l_{P_{k+1},1}$) of the $(k + 1)$ -st while-loop, (93) still holds. When we start the $(k + 1)$ -st while-loop, we have $\text{count} = 1$, and $r(l_{P_{k+1},1})$ must be in one of the following two cases: 1) $r(l_{P_{k+1},1}) = -1$ if the rank of link $l_{P_{k+1},1}$ is not assigned yet, or 2) $r(l_{P_{k+1},1}) \geq \text{count}$, otherwise. Then, Algorithm 2 will assign a rank of 1 to link $l_{P_{k+1},1}$ in the former case (line 9), or will not change its rank in the latter case (line 11). Hence, (93) still holds.

Now suppose that after assigning the ranks of links up to link $l_{P_{k+1},n}$, which is the n -th hop of the flow-path chosen in the $(k+1)$ -st while-loop, we have $r(l_{P_{k+1},m-1}) < r(l_{P_{k+1},m})$ for all $1 < m \leq n$, and (93) holds. Then we want to show that after assigning a rank to the next hop $l_{P_{k+1},n+1}$, we still have both $r(l_{P_{k+1},m-1}) < r(l_{P_{k+1},m})$ for all $m \leq n+1$, and (93). We show this when $n = 2$ for ease of presentation. One can easily extend the analysis to the case when $n \geq 2$. After assigning a rank to link $l_{P_{k+1},1}$, we have $count = r(l_{P_{k+1},1}) + 1$ from line 18 of Algorithm 2. At this moment, the rank of link $l_{P_{k+1},2}$ is either 1) $r(l_{P_{k+1},2}) = -1$, 2) $r(l_{P_{k+1},2}) \geq count$, or 3) $0 < r(l_{P_{k+1},2}) < count$. We discuss the three cases as follows.

Case 1): $r(l_{P_{k+1},2}) = -1$.

In this case, since Algorithm 2 sets $r(l_{P_{k+1},2})$ to $count$ from line 9, we have $r(l_{P_{k+1},2}) > r(l_{P_{k+1},1})$. The rank of links of P_j for all $j \leq k$ is not changed, and (93) still holds.

Case 2): $r(l_{P_{k+1},2}) \geq count$.

In this case, since Algorithm 2 does not change the rank $r(l_{P_{k+1},2})$, we have $r(l_{P_{k+1},2}) \geq count > r(l_{P_{k+1},1})$. The rank of links of P_j for all $j \leq k$ is not changed, and (93) still holds.

Case 3): $0 < r(l_{P_{k+1},2}) < count$.

Note that in this case, we have $r(l_{P_{k+1},1}) \geq r(l_{P_{k+1},2})$ before assigning a new rank to link $l_{P_{k+1},2}$. Since Algorithm 2 sets $r(l_{P_{k+1},2})$ to $count$ in line 16, we will have $r(l_{P_{k+1},2}) > r(l_{P_{k+1},1}) = count - 1$. Now what remains to show is that after the rank update for links of $\Gamma_k(l_{P_{k+1},2})$ in lines 13-15, we still have $r(l_{P_{k+1},2}) > r(l_{P_{k+1},1})$ and (93) still holds.

Recall that $\Gamma_k(l) = \{l' \in \bigcup_{P \in \mathcal{P}(l) \cap \mathcal{P}_k} \mathcal{E}(P) \mid r(l') > r(l)\}$ denotes the set of links that belong to the flow-paths of $\mathcal{P}(l) \cap \mathcal{P}_k(\mathcal{T})$ (i.e., flow-paths that pass through link l and are chosen in the j -th while-loop for $j < k$) and have a rank greater than $r(l)$. Let $\Omega \triangleq \Gamma_{k+1}(l_{P_{k+1},2}) \cup \{l_{P_{k+1},2}\}$ denote the union of $\Gamma_{k+1}(l_{P_{k+1},2})$ and $\{l_{P_{k+1},2}\}$. Algorithm 2 updates only the rank of the links in Ω by adding the rank with $count - r(l_{P_{k+1},2})$. We claim that $l_{P_{k+1},1} \notin \Omega$, i.e., the rank $r(l_{P_{k+1},1})$ is not changed after the update, which implies that $r(l_{P_{k+1},2}) > r(l_{P_{k+1},1})$ still holds after the update. We prove this claim by contradiction. Suppose that $l_{P_{k+1},1} \in \Omega$, then there exists a flow-path $P' \in \mathcal{P}(l_{P_{k+1},2}) \cap \mathcal{P}_{k+1}$ such that $l_{P_{k+1},1}, l_{P_{k+1},2} \in \mathcal{E}(P')$ and link $l_{P_{k+1},2}$ appears earlier than $l_{P_{k+1},1}$ on the flow-path P' . This implies that flow-paths P' and P_{k+1} form a flow-loop, which contradicts with the definition of flow-tree.

Next, we want to show that (93) still holds after the rank update. Note that before the rank update, due to (93), two adjacent links $l_{P_j,i}$ and $l_{P_j,i+1}$ satisfy that $r(l_{P_j,i}) < r(l_{P_j,i+1})$ for any $j \leq k$ and any $i < len(P_j)$. We want to show that, after the rank update, we still have $r(l_{P_j,i}) < r(l_{P_j,i+1})$. We consider the following four cases for two adjacent links $l_{P_j,i}$ and $l_{P_j,i+1}$.

Case i): $l_{P_j,i} \in \Omega$ and $l_{P_j,i+1} \in \Omega$.

In this case, since Algorithm 2 increases the rank of links $l_{P_j,i}$ and $l_{P_j,i+1}$ by $\text{count} - r(l_{P_{k+1},2})$, we still have $r(l_{P_j,i}) < r(l_{P_j,i+1})$ after the update.

Case ii): $l_{P_j,i} \notin \Omega$ and $l_{P_j,i+1} \notin \Omega$.

In this case, since Algorithm 2 does not change the rank of links $l_{P_j,i}$ and $l_{P_j,i+1}$, we still have $r(l_{P_j,i}) < r(l_{P_j,i+1})$ after the update.

Case iii): $l_{P_j,i} \notin \Omega$ and $l_{P_j,i+1} \in \Omega$.

In this case, since Algorithm 2 increases the rank of link $l_{P_j,i+1}$ by $\text{count} - r(l_{P_{k+1},2})$ and does not change the rank of links $l_{P_j,i}$, we still have $r(l_{P_j,i}) < r(l_{P_j,i+1})$ after the update.

Case iv): $l_{P_j,i} \in \Omega$ and $l_{P_j,i+1} \notin \Omega$.

This is an infeasible case from the definition of Ω and (93) of the previous step. Note that since links $l_{P_j,i}$ and $l_{P_j,i+1}$ are two adjacent links on the flow-path P_j , there exists a flow s such that $l_{P_j,i}, l_{P_j,i+1} \in \mathcal{L}(s)$ from the definition of flow-path (Definition 5), we should have $r(l_{P_j,i}) < r(l_{P_j,i+1})$ before the rank update. Hence if $l_{P_j,i} \in \Omega$, we should have $l_{P_j,i+1} \in \Omega$ from the definition of Ω .

We can show the property of monotonically increasing ranking for Case 3) by combining sub-cases i), ii), iii) and iv). Results for Cases 1), 2) and 3) complete the induction step when $n = 2$. One can easily extend the analysis to the case when $n \geq 2$, and this completes the proof.

APPENDIX H

PROOF OF PROPOSITION 4

We want to show that, a network where flows do not form loops, i.e., all the components are flow-trees, is stable under FLQ-MWS for any traffic with arrival rate vector that is strictly inside Λ^* .

We know from Lemma 3 that, there exists a ranking $R(\mathcal{E})$ such that the monotone property holds. Without loss of generality, we assume that the minimum rank is 1, and use $r(\mathcal{E}) \triangleq \max_{l \in \mathcal{E}} r(l)$ to denote the maximum rank among all the links. We give the following definitions that are used in the proof.

Definition 6: We divide \mathcal{E} into $r(\mathcal{E})$ disjoint subsets: $R_k \triangleq \{l \in \mathcal{E} \mid r(l) = k\}$, for $1 \leq k \leq r(\mathcal{E})$. Then R_k is called the *depth-k set*, and a link $l_k \in R_k$ is called a *depth-k link*.

Recall that the fluid limit model for the sub-system consisting of shadow queues is stable from Lemma 20. We show by induction that all data queues are stable.

Base Case:

First, Lemma 3 implies that for any $l_1 \in R_1$, its arrivals are exogenous, i.e., $A_{l_1}(t) = \sum_s H_{l_1,1}^s F_s(t)$. Following the same line of analysis for the proof of Proposition 1, we can show that $\pi_{l_1}(t) \geq (1 +$

$\epsilon) \sum_s H_{l_1,1}^s \lambda_s$ and $p_{l_1}(t) = \sum_s H_{l_1,1}^s \lambda_s$, then $\frac{d}{dt} q_{l_1}(t) = p_{l_1}(t) - \pi_{l_1}(t) \leq -\epsilon \sum_s H_{l_1,1}^s \lambda_s < 0$, if $q_{l_1}(t) > 0$. This implies that $q_{l_1}(t)$ is stable, for all $l_1 \in R_1$.

Induction Step:

Next, we show that, if q_l is stable for all $l \in \bigcup_{j \leq k} R_j$, then $q_{l_{k+1}}$ is also stable for all $l_{k+1} \in R_{k+1}$, along with the stability of all q_l , for $1 \leq k < K$.

Lemma 3 implies that for any $l_{k+1} \in R_{k+1}$, its arrivals are either exogenous or from certain links of $\bigcup_{j \leq k} R_j$. Since q_l is stable for all $l \in \bigcup_{j \leq k} R_j$, following the same line of analysis for the proof of Proposition 1, we can show that there exists a finite time $T_3^k > 0$ such that, for all time $t \geq T_3^k$, we have $\pi_{l_{k+1}}(t) \geq (1 + \epsilon) \sum_{s: l_{k+1} \in \mathcal{L}(s)} \lambda_s$ and $p_{l_{k+1}}(t) = \sum_{s: l_{k+1} \in \mathcal{L}(s)} \lambda_s$. Therefore, for all time $t \geq T_3^k$, we have $\frac{d}{dt} q_{l_{k+1}}(t) = p_{l_{k+1}}(t) - \pi_{l_{k+1}}(t) \leq -\epsilon \sum_{s: l_{k+1} \in \mathcal{L}(s)} \lambda_s < 0$, if $q_{l_{k+1}}(t) > 0$. This implies that $q_{l_{k+1}}$ is stable for all $l_{k+1} \in R_{k+1}$.

Therefore, the fluid limit model for the sub-system of data queues is stable from the induction. With Lemma 20, this implies that the fluid limit model of the joint system of data queues and shadow queues is stable. Then, we complete the proof following the same arguments used in the proof of Proposition 1.

APPENDIX I

PROOF OF LEMMA 7

Given any $\gamma \in (0, 1)$, suppose that λ is strictly inside $(1 - \gamma)\Lambda^*$, then there exists a sufficiently small $\epsilon > 0$ such that $(1 + \epsilon)\lambda$ is strictly inside $(1 - \gamma)\Lambda^*$, and we can find a vector $\phi \in (1 - \gamma)Co(\mathcal{M})$ such that $(1 + \epsilon)\lambda < \phi$, i.e., $(1 + \epsilon) \sum_s \sum_k H_{l,k}^s \lambda_s < \phi_l$ for all $l \in \mathcal{E}$. Let $\beta \triangleq \min_{l \in \mathcal{E}} (\phi_l - (1 + \epsilon) \sum_s \sum_k H_{l,k}^s \lambda_s)$. By definition, we have $\beta > 0$. Let T' be a finite time such that $T' > \frac{(1 + \epsilon)}{\beta}$. Then, for all regular time $t \geq T'$, we have

$$\hat{p}_l(t) \leq (1 + \epsilon) \left(\sum_s \sum_k H_{l,k}^s \lambda_s + \frac{1}{t} \right) < \phi_l, \quad (94)$$

from Lemma 19. This implies that the instantaneous arrival rate of shadow queues is strictly inside $(1 - \gamma)$ fraction of the optimal throughput region Λ^* .

Let $W_l(\hat{q}_l) \triangleq \int_0^{\hat{q}_l} g_l(y) dy$ and consider a Lyapunov function $\hat{V}(\hat{q}(t)) \triangleq \sum_l W_l(\hat{q}_l(t))$. It is sufficient to show that for any $\zeta_1 > 0$, there exists a $\zeta_2 > 0$ such that $\hat{V}(\hat{q}(t)) \geq \zeta_1$ implies $\frac{D^+}{dt} \hat{V}(\hat{q}(t)) \leq -\zeta_2$, for any regular time $t \geq T'$. Since $W_l(\hat{q}_l)$'s and \hat{q}_l 's are differentiable, for any regular time $t \geq T'$, we

can obtain the derivative of $\hat{V}(\hat{q}(t))$ as

$$\begin{aligned} \frac{D^+}{dt^+} \hat{V}(\hat{q}(t)) &= \sum_{l \in \mathcal{E}} g_l(\hat{q}_l(t)) \cdot \frac{d}{dt} \hat{q}_l(t) = \sum_{l \in \mathcal{E}} g_l(\hat{q}_l(t)) \cdot (\hat{p}_l(t) - \hat{\pi}_l(t)) \\ &= \sum_{l \in \mathcal{E}} g_l(\hat{q}_l(t)) \cdot (\hat{p}_l(t) - \phi_l) + \sum_{l \in \mathcal{E}} g_l(\hat{q}_l(t)) \cdot (\phi_l - \hat{\pi}_l(t)). \end{aligned} \quad (95)$$

Let us choose $\zeta_3 > 0$ such that $\hat{V}(\hat{q}(t)) \geq \zeta_1$ implies $\max_l \hat{q}_l(t) \geq \zeta_3$. Then following a similar argument as in the proof of Lemma 15, for the final result of (55), we can conclude that the first term is bounded as follows:

$$\sum_{l \in \mathcal{E}} g_l(\hat{q}_l(t)) \cdot (\hat{p}_l(t) - \phi_l) \leq -\zeta_2 < 0,$$

and that the second term becomes non-positive due to the following. We first note that $\|\hat{q}(t)\| > 0$ from $\hat{V}(\hat{q}(t)) > 0$. Then at time slots $\Upsilon \triangleq \{[x_{n_j} t], [x_{n_j} t] + 1, \dots, [x_{n_j} (t + \delta)]\}$, for any $Q_B > 0$, we have $\|\hat{Q}(\tau)\| \geq Q_B$ for all time slots $\tau \in \Upsilon$ with large enough j and small enough δ . From Lemma 6, given any $\theta \in (0, 1)$, for all time slots $\tau \in \Upsilon$, with probability greater than $1 - \theta$, LQ-CSMA chooses a schedule $M(\tau) \in \mathcal{M}$ that satisfies

$$\sum_{l \in \mathcal{E}} g_l(\hat{Q}_l(\tau)) \cdot M_l(\tau) \geq (1 - \gamma) \max_{M \in \mathcal{M}} \sum_{l \in \mathcal{E}} g_l(\hat{Q}_l(\tau)) \cdot M_l. \quad (96)$$

Hence, similar as in Chapter 4 of [24], from condition (9), with probability greater than $1 - \theta$, the fluid limit $\hat{\pi}(t)$ under LQ-CSMA satisfies

$$\begin{aligned} \sum_{l \in \mathcal{E}} g_l(\hat{q}_l(t)) \cdot \hat{\pi}_l(t) &\geq (1 - \gamma) \max_{\phi' \in Co(\mathcal{M})} \sum_{l \in \mathcal{E}} g_l(\hat{q}_l(t)) \cdot \phi'_l \\ &= \max_{\phi \in (1-\gamma)Co(\mathcal{M})} \sum_{l \in \mathcal{E}} g_l(\hat{q}_l(t)) \cdot \phi_l. \end{aligned} \quad (97)$$

Therefore, $\hat{V}(\hat{q}(t)) \geq \zeta_1$ implies $\frac{D^+}{dt^+} \hat{V}(\hat{q}(t)) \leq -\zeta_2$. This completes the proof.

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] G. Sharma, R. R. Mazumdar, and N. B. Shroff, “On the complexity of scheduling in wireless networks,” in *Proceedings of the annual international conference on Mobile computing and networking (MobiCom)*. ACM New York, NY, USA, 2006, pp. 227–238.
- [3] L. Jiang and J. Walrand, “A distributed CSMA algorithm for throughput and utility maximization in wireless networks,” *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 960–972, 2010.
- [4] 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,” *Arxiv preprint arXiv:0901.2333*, 2009. [Online]. Available: http://arxiv.org/PS_cache/arxiv/pdf/0901/0901.2333v4.pdf

- [5] S. Rajagopalan, D. Shah, and J. Shin, "Network adiabatic theorem: an efficient randomized protocol for contention resolution," in *The ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, 2009, pp. 133–144.
- [6] L. Bui, R. Srikant, and A. Stolyar, "A Novel Architecture for Reduction of Delay and Queueing Structure Complexity in the Back-Pressure Algorithm," *IEEE/ACM Transactions on Networking*, vol. 19, no. 6, pp. 1597–1609, December 2011.
- [7] A. Stolyar, "Large number of queues in tandem: Scaling properties under back-pressure algorithm," *Queueing Systems*, vol. 67, no. 2, pp. 111–126, 2011.
- [8] S. Liu, E. Ekici, and L. Ying, "Scheduling in Multihop Wireless Networks without Back-pressure," in *Proceedings of the Annual Conference on Communication, Control and Computing (Allerton)*, 2010.
- [9] L. Ying, R. Srikant, and D. Towsley, "Cluster-based back-pressure routing algorithm," in *The IEEE International Conference on Computer Communications (INFOCOM)*, 2008, pp. 484–492.
- [10] L. Ying, S. Shakkottai, and A. Reddy, "On combining shortest-path and back-pressure routing over multihop wireless networks," in *The IEEE International Conference on Computer Communications (INFOCOM)*, 2009, pp. 1674–1682.
- [11] X. Wu, R. Srikant, and J. Perkins, "Scheduling Efficiency of Distributed Greedy Scheduling Algorithms in Wireless Networks," *IEEE Transactions on Mobile Computing*, pp. 595–605, 2007.
- [12] IETF, "Rfc 2453," 1998.
- [13] ———, "Rfc 791," 1981.
- [14] "Internet world stats," March 2011. [Online]. Available: <http://www.internetworldstats.com/stats.htm>
- [15] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting, *Scheduling in a queueing system with asynchronously varying service rates*. Cambridge Univ Press, 2004, vol. 18.
- [16] M. Bramson, "Stability of queueing networks," *Probability Surveys*, vol. 5, no. 1, pp. 169–345, 2008.
- [17] J. Dai, "On positive Harris recurrence of multiclass queueing networks: a unified approach via fluid limit models," *The Annals of Applied Probability*, pp. 49–77, 1995.
- [18] A. Rybko and A. Stolyar, "Ergodicity of stochastic processes describing the operation of open queueing networks," *Problems of Information Transmission*, vol. 28, pp. 199–220, 1992.
- [19] S. Lu and P. Kumar, "Distributed scheduling based on due dates and buffer priorities," *IEEE Transactions on Automatic Control*, vol. 36, no. 12, pp. 1406–1416, 1991.
- [20] H. Chen and H. Zhang, "Stability of multiclass queueing networks under priority service disciplines," *Operations Research*, pp. 26–37, 2000.
- [21] H. Chen and H. Ye, "Piecewise linear Lyapunov function for the stability of multiclass priority fluid networks," *IEEE Transactions on Automatic Control*, vol. 47, no. 4, pp. 564–575, 2002.
- [22] H. Chen and D. Yao, "Stable priority disciplines for multiclass networks," in *Stochastic Networks: Stability and Rare Events*, P. Glasserman, K. Sigman, and D. Yao, Eds. Springer-Verlag, 1996, ch. 2, pp. 27–39.
- [23] A. Eryilmaz, R. Srikant, and J. Perkins, "Stable scheduling policies for fading wireless channels," *IEEE/ACM Transactions on Networking*, vol. 13, no. 2, pp. 411–424, 2005.
- [24] D. Shah, "Randomization and heavy traffic theory: new approaches to the design and analysis of switch algorithms," Ph.D. dissertation, Stanford University, 2004.
- [25] B. Ji, C. Joo, and N. B. Shroff, "Delay-Based Back-Pressure Scheduling in Multi-Hop Wireless Networks," in *The IEEE International Conference on Computer Communications (INFOCOM)*, 2011, pp. 2579–2587.

- [26] V. Malyshev and M. Menshikov, "Ergodicity, continuity and analyticity of countable Markov chains," *Transactions of the Moscow Mathematical Society*, vol. 39, pp. 3–48, 1979.
- [27] J. Dai and B. Prabhakar, "The throughput of data switches with and without speedup," in *The IEEE International Conference on Computer Communications (INFOCOM)*, 2000, pp. 556–564.