# Flow-Level Stability of Wireless Networks: Separation of Congestion Control and Scheduling

J. Ghaderi, *Student Member, IEEE*, T. Ji, *Student Member, IEEE*, R. Srikant, *Fellow, IEEE*

***Abstract*—It is by now well-known that wireless networks with file arrivals and departures are stable if one uses $\alpha$-fair congestion control and back-pressure based scheduling and routing. In this paper, we examine whether $\alpha$-fair congestion control is necessary for flow-level stability. We show that stability can be ensured even with very simple congestion control mechanisms, such as a fixed window size scheme which limits the maximum number of packets that are allowed into the ingress queue of a flow. A key ingredient of our result is the use of the difference between the logarithms of queue lengths as the link weights. This result is reminiscent of results in the context of CSMA algorithms, but for entirely different reasons.**

## I. Introduction

In order to operate wireless systems efficiently, scheduling algorithms are needed to facilitate simultaneous transmissions of different users. Scheduling algorithms for wireless networks have been widely studied since Tassiulas and Ephremides [1] proposed the *max weight* algorithm for single-hop wireless networks and its extension to multihop networks using the notion of *back-pressure* or *differential backlog*. Such algorithms assign a weight to each link as a function of the number of packets queued at the link, and then, at each instant of time, select the schedule with the maximum weight, where the weight of a schedule is computed by summing the weights of the links that the schedule will serve. Tassiulas and Ephremides establish that the back-pressure algorithm (and hence, the max weight algorithm) is throughput optimal in the sense that it can stabilize the queues of the network for the largest set of arrival rates possible *without actually knowing the arrival rates*. The back-pressure algorithm works under very general conditions but it does not consider *flow-level dynamics*. It considers *packet-level dynamics* assuming that there is a fixed set of users/flows and packets are generated by each flow according to some stochastic process. In real networks however, flows arrive randomly to the network, have only a finite amount of data, and depart the network after the data transfer is completed. Moreover, there is no notion of *congestion control* in the back-pressure algorithm while most modern communication networks use some congestion control mechanism for fairness purposes or to avoid excessive congestion inside the network [2].

There is a rich body of literature on the packet-level stability of scheduling algorithms, e.g., [1], [7], [8], [9]. Stability of wireless networks under flow-level dynamics has been studied in, e.g., [2], [3], [4]. Here, by stability, we mean that the number of flows in the network and the queue sizes at each node in the network remain finite.

Under flow-level dynamics, if the scheduler has access to the total queue-length information at nodes, then it can use max weight/back-pressure algorithm to achieve throughput optimality, but this information is not typically available to the scheduler because it is implemented as part of the MAC layer. Moreover, without congestion control, queue sizes at different nodes could be widely different. This could lead to long periods of unfairness among flows because links with long flows/files (large weights) will get priority over links with short flows/files (small weights) for long periods of time. Therefore, we need to use congestion control to provide better *Quality-of-Service* (QoS). With congestion control, only a few packets from each file are released to the MAC layer at each time instant, and scheduling is done based on these MAC layer packets. Specifically, the network control policy consists of two parts: (a) "congestion control" which determines the rate of service provided to each flow, and (b) "packet scheduling" which determines the rate of service provided to each link in the network.

However, to achieve flow-level stability, prior works [2], [3], [4] require that a specific form of congestion control has to be used, namely, *ingress queue based rate adaptation using $\alpha$-fair utility functions*. More accurately, (a) the rate at which a flow/file generates packets into its ingress queue must maximize its utility subject to a linear penalty (price). The utility function of each flow is assumed to be in the form $x^{1-\alpha}/(1-\alpha)$, for some $\alpha > 0$, with $x$ the flow rate, and the penalty (price) charged is the number of packets queued at the ingress queue associated with the flow, (b) scheduling of packets is performed using the max weight/back-pressure algorithm, where the weight of each link is the queue size (or the queue size raised to the power $\alpha$).

In this paper, we show that $\alpha$-fair congestion control *is not* necessary for flow-level stability, and, in fact, very general congestion control mechanisms are sufficient to ensure flow-level stability. The result suggests that ingress queue-based congestion control is more important than $\alpha$-fairness to ensure network stability, when congestion control is used in conjunction with max weight scheduling/routing. As an example, a simple fixed window size scheme which limits the maximum number of packets allowed into the ingress queue of each flow can provide flow-level stability.

In establishing the above result, we have used the max weight algorithm with link weights which are *log-differentials* of MAC-layer queue lengths, i.e., the weight of a link $(i, j)$ is

The authors are with the Department of Electrical and Computer Engineering, and the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA (emails:{jghaderi, tji2, rsrikant}@illinois.edu).

The research was supported in part by ARO MURIs W911NF-07-1-0287 and W911NF-08-1-0233 and AFOSR MURI FA 9550-10-1-0573.

chosen to be in the form of $\log(1+q_i)-\log(1+q_j)$, where $q_i$ and $q_j$ are the MAC-layer queue lengths of nodes $i$ and $j$. Such a choice of link weights is crucial in establishing our stability result. Note that we only use MAC-layer queue lengths that are readily available at the nodes and do not involve knowing the number of existing files at Transport layers.

The use of logarithmic functions of queue lengths naturally suggests the use of a CSMA *(Carrier Sense Multiple Access)*-type algorithm to implement the scheduling algorithm in a distributed fashion [12], [13], [10]. The main difference here is that the weights are $\log$-differential of queue lengths rather than $\log$ of queue lengths themselves, and thus results in [12], [13], [10] are not directly applicable. We show that the stability results for CSMA without time-scale separation can be extended to the multihop model in this paper with log-differential of queue lengths as weights, and the type of congestion control mechanism considered here.

The main contributions of this paper can be summarized as follows:

1) We show that $\alpha$-fair congestion control is not necessary for stability, and, in fact, very general ingress queue based congestion control mechanisms are sufficient to ensure stability. A key ingredient of our result is the use of the difference between the logarithms of queue lengths as the link weights.
2) The design of efficient scheduling and congestion control algorithms can be decoupled. This separation result would allow using different congestion control mechanisms at the edge of the network for providing different fairness or QoS considerations without need to change the scheduling algorithm implemented at internal routers of the network.
3) A by-product of the weight function that we use for each link is that one can use CSMA to implement the scheduling algorithm in a distributed fashion. In particular, unlike [5] which also considers flow-level stability, we do not have to assume time-scale separation between the dynamics of flows, packets, and CSMA algorithm.

We note that earlier versions of this paper appeared in [21], [23] without many of the details and the proofs included here. In particular, all three time-scales (flow, packet, and MAC algorithm) are considered in the analytical results here unlike the earlier versions. One interesting aspect of the simulations in [21] is that it suggests that the advantage of using only MAC-layer queue length for scheduling is that it reduces delays dramatically for short flows. However, due to space limitations, simulation results are not presented here, but the interested reader can find them in [21].

The rest of the paper is organized as follows. In Section II, we describe our models for the wireless network, file arrivals, and Transport and MAC layers. We propose our scheduling algorithm in Section III. Section IV is devoted to the formal statement about the throughput-optimality of the algorithm and its proof. In Section V, we consider the distributed implementation of our algorithm. Section VI contains conclusions and possible future research directions. The appendices at the end of the paper contain some of the proofs.

## II. System Model

### *Wireless network model*

Consider a multihop wireless network consisting of a set of nodes $\mathcal{N} = \{1, 2, .., N\}$ and a set of links $\mathcal{L}$ between the nodes. There is a link from $i$ to $j$, i.e., $(i, j) \in \mathcal{L}$, if transmission from $i$ to $j$ is allowed. There is a set of users/source nodes $\mathcal{U} \subseteq \mathcal{N}$ and each user/source transfers data to a destination over a fixed route in the network[1]. For a user/source $u \in \mathcal{U}$, we use $d(u)(\neq u)$ to denote its destination. Let $\mathcal{D} := d(\mathcal{U})$ denote the set of all destinations.

We consider a time-slotted system. At each time slot $t$, files of different sizes arrive at the source nodes. As in the standard congestion control algorithm, TCP, files inject packets into their MAC-layer queues. The packets then travel to their respective destinations in a multihop fashion, i.e., along links in the network with queueing in buffers at intermediate nodes. Transmission of each packet along its route is subject to physical layer constraints such as interference and limited link capacity.

Let $\mathcal{R}$ denote the set of available rate vectors (or transmission schedules) $\mathbf{r} = [r_{ij} : (i, j) \in \mathcal{L}]$ at each time slot. Thus, $r_{ij}$ is the number of packets that can be transmitted from $i$ to $j$ during time slot $t$ if the transmission schedule $\mathbf{r}$ is selected at time slot $t$. Note that each transmission schedule $\mathbf{r}$ corresponds to a set of node power assignments chosen by the network. Also let $\text{Co}(\mathcal{R})$ denote the convex hull of $\mathcal{R}$ which corresponds to time-sharing between different rate vectors. Hence, if $\gamma = [\gamma_{ij} : (i, j) \in \mathcal{L}]$ denotes the average rate of service provided to the links, then in general, $\gamma \in \text{Co}(\mathcal{R})$.

### *Traffic model*

We use $a_s(t)$ to denote the number of files that arrive at source $s$ at time $t$ and assume that the process $\{a_s(t); s \in \mathcal{U}\}_{\{t=1,2,\cdots\}}$ is i.i.d. over time and independent across users with rate $[\kappa_s; s \in \mathcal{U}]$ and has bounded second moments. Moreover, we assume that there are $K$ possible file types where the files of type $i$ are geometrically distributed with mean $1/\eta_i$ packets. The file arrived at source $s$ can belong to type $i$ with probability $p_{si}$, $i = 1, 2, .., K$, $p_{si} \geq 0$, $\sum_{i=1}^{k} p_{si} = 1$. Our motivation for selecting such a model is due to the *large variance distribution* of file sizes in the Internet. It is believed, see e.g., [15], that most of bytes are generated by long files while most of the files are short files. By controlling the probabilities $p_{si}$, for the same average file size, we can obtain distributions with very large variance. Let $m_s := \sum_{i=1}^{K} p_{si}/\eta_i$ denote the mean file size at node $s$, and define the work load at source $s$ by $\rho_s := \kappa_s m_s$. Let $\boldsymbol{\rho} := [\rho_s : s \in \mathcal{U}]$ be the vector of such work loads in the network .

### *Transport and MAC layers*

Upon arrival of a file at a source Transport layer, a TCP-connection is established that regulates the injection of packets

[1]The final results can be extended to case when each source has multiple destinations or to the cases of multi-path routing and adaptive routing. Here, to expose the main features, we have considered a simpler model.

into the MAC layer. Once transmission of a file ends, the file departs and the corresponding TCP-connection will be closed. The MAC-layer is responsible for making the scheduling decisions to deliver the MAC-layer packets to their destinations over their corresponding routes. Each node has a fixed routing table that determines the next hop for each destination.

At each source node, we index the files according to their arriving order such that the index 1 is given to the earliest file. This means that once transmission of a file ends, the indices of the remaining files are updated such that indices again start from 1 and are consecutive. Note that the indexing rule *is not* part of the algorithm implementation and it is used here only for the purpose of analysis. We use $\mathcal{W}_{sf}(t)$ to denote the TCP *congestion window size* for file $f$ at source $s$ at time $t$. Hence, $\mathcal{W}_{sf}$ is a time-varying sequence which changes as a result of TCP congestion control. If the congestion window of file $f$ is not full, TCP will continue injecting packets from the remainder of file $f$ to the congestion window until file $f$ has no packets remaining at the Transport layer or the congestion window becomes full. We consider *ingress queue-based congestion control* meaning that when a packet of congestion window departs the ingress queue, it is replaced with a new packet from its corresponding file at the Transport layer. It is important to note that the MAC layer does not know the number of remaining packets at the Transport layer, so scheduling decisions have to be made based on the MAC-layers information only. It is reasonable to assume that $1 \leq \mathcal{W}_{sf}(t) \leq \mathcal{W}_{cong}$, i.e., each file has at least one packet waiting to be transferred and all congestion window sizes are bounded from above by a constant $\mathcal{W}_{cong}$.

*Routing and queue dynamics*

At the MAC layer of each node $n \in \mathcal{N}$, we consider separate queues for the packets of different destinations. Let $q_n^{(d)}$, $d \in \mathcal{D}$, denote the packets of destination $d$ at the MAC-layer of $n$. Also let $\mathbf{R}^{(d)}_{N\times N}$ be the routing matrix corresponding to packets of destination $d$ where $R_{ij}^{(d)} = 1$ if the next hop of node $i$ for destination $d$ is node $j$, for some $j$ such that $(i,j) \in \mathcal{L}$, and 0 otherwise. Routes are *acyclic* meaning that each packet eventually reaches its destination and leaves the network. A packet of destination $d$ that is transmitted from $i$ to $j$ is removed from $q_i^{(d)}$ and added to $q_j^{(d)}$. Packet that reaches its destination is removed from the network. Note that packets in $q_n^{(d)}$ could be generated at node $n$ itself (if $n$ is a source with destination $d$) or belong to other sources that use $n$ as an intermediate relay along their routes to destination $d$.

## III. Description of Scheduling Algorithm

The algorithm is essentially the back-pressure algorithm [1] but it only uses the MAC-layer information. The key step in establishing the optimality of such an algorithm is using an appropriate weight function of the MAC-layer queues instead of using the total queues. In particular, consider a *log-type* function

$$g(x) := \frac{\log(1+x)}{h(x)}, \tag{1}$$

where $h(x)$ is an arbitrary increasing function which makes $g(x)$ an increasing concave function. Assume that $h(0) > 0$ and $g(x)$ is continuously differentiable on $(0, \infty)$: For example, $h(x) = \log(e + \log(1+x))$ or $h(x) = \log^{\theta}(e+x)$ for some $0 < \theta < 1$. For each link $(i,j)$ with $R_{ij}^{(d)} = 1$, define

$$w_{ij}^{(d)}(t) := g\Big(q_i^{(d)}(t)\Big) - g\Big(q_j^{(d)}(t)\Big). \tag{2}$$

Note that if $\{d \in \mathcal{D} : R_{ij}^{(d)} = 1\} = \emptyset$, then we can remove the link $(i,j)$ from the network without reducing the capacity region since no packets are forwarded over it. So without loss of generality, we assume that $\{d \in \mathcal{D} : R_{ij}^{(d)} = 1\} \neq \emptyset$, for every $(i,j) \in \mathcal{L}$.

Let $x_{ij}^{(d)}(t)$ denote the scheduling variable that shows the rate at which the packets of destination $d$ can be forwarded over the link $(i,j)$ at time slot $t$. The scheduling algorithm is as follows.

At each time $t$:

- Each node $n$ observes the MAC-layer queue sizes of itself and its next hop, i.e., for each $d \in \mathcal{D}$, it observes $q_n^{(d)}$ and $q_j^{(d)}$ for a $j$ such that $R_{ij}^{(d)} = 1$.
- For each link $(i,j)$, calculate a weight

$$w_{ij}(t) := \max_{d \in \mathcal{D}: R_{ij}^{(d)}=1} w_{ij}^{(d)}(t), \tag{3}$$

and

$$\tilde{d}_{ij}^{*}(t) := \arg\max_{d \in \mathcal{D}: R_{ij}^{(d)}=1} w_{ij}^{d}(t). \tag{4}$$

- Find the optimal rate vector $\tilde{x}^* \in \mathcal{R}$ that solves

$$\tilde{x}^*(t) = \arg\max_{r \in \mathcal{R}} \sum_{(i,j)\in\mathcal{L}} r_{ij} w_{ij}(t). \tag{5}$$

- Finally, assign $x_{ij}^{(d)}(t) = \tilde{x}_{ij}^{*}$ if $d = \tilde{d}_{ij}^{*}(t)$, and zero otherwise (break ties at random).

## IV. System Stability

In this section, we analyze the system and prove its stability under the algorithm described in Section III.

For the analysis, we use $Q_n^{(d)}$ (*with capital Q*) to denote the *total per-destination queues*, i.e., the total number of packets of destination $d$ at node $n$, in its MAC or Transport layer. Note that, for each node $n$, the MAC (or total) per-destination queues $q_n^{(d)}$ (or $Q_n^{(d)}$) fall into three cases: (i) $n$ is source and $d$ is its destination, (ii) $n$ is a source but $d$ is not its destination, and (iii) $n$ is not a source. In the case (i), it is important to distinguish between the MAC-layer queue and the total queue associated with $d$, i.e., $Q_n^{(d)} \neq q_n^{(d)}$, because of the existing packets of destination $d$ at the Transport layer of $n$. However, $Q_n^{(d)} = q_n^{(d)}$ holds in case (ii), and for all destinations in case (iii).

Let $z_{ij}(t)$ denote the number of packets transmitted over link $(i,j) \in \mathcal{L}$ at time $t$. Then, the total-queue dynamics for a destination $d$, at each node $n$, is given by

$$Q_n^{(d)}(t+1) = Q_n^{(d)}(t) - \sum_{j=1}^{N} R_{nj}^{(d)} z_{nj}^{(d)}(t) + \sum_{i=1}^{N} R_{in}^{(d)} z_{in}^{(d)}(t) + A_n^{(d)}(t), \quad (6)$$

where $A_n^{(d)}(t)$ is the total number of packets for destination $d$ that new files bring to node $n$ at time slot $t$. Note that $A_n^{(d)}(t) \equiv 0$ in the cases (ii) and (iii) above. With minor abuse of notation, we write $\mathbb{E}\left[A_n^{(d)}(t)\right] = \rho_n^{(d)}$ with $\rho_n^{(d)} := \rho_n$ in the case (i) and $\rho_n^{(d)} := 0$ otherwise. Also $z_{ij}^{(d)}(t) = \min\left\{x_{ij}^{(d)}(t), q_i^{(d)}(t)\right\}$ obviously, because $i$ cannot send more than its MAC-layer queue content at each time.

**Definition 1.** *The capacity region of the network $\mathcal{C}$ is defined as the set of all load vectors $\boldsymbol{\rho}$ that under which the total queues in the network can be stabilized. Note that under our flow-level model, stability of total queues will imply that the number of files in the network is also stable. It is well-known, see e.g. [7], that a vector $\boldsymbol{\rho}$ belongs to $\mathcal{C}$ if and only if there exits an average service rate vector $\gamma \in Co(\mathcal{R})$ such that*

$$\gamma_{ij}^{(d)} \geq 0;\ \forall d \in \mathcal{D} \text{ and } \forall (i,j) \in \mathcal{L},$$
$$\rho_n^{(d)} - \sum_{j=1}^{N} R_{nj}^{(d)} \gamma_{nj}^{(d)} + \sum_{i=1}^{N} R_{in}^{(d)} \gamma_{in}^{(d)} \leq 0;\ \forall d \in \mathcal{D},\ \forall n \neq d,$$
$$\sum_{d \in \mathcal{D}} \gamma_{ij}^{(d)} \leq \gamma_{ij};\ \forall (i,j) \in \mathcal{L}.$$

**Theorem 1.** *For any $\boldsymbol{\rho}$ strictly inside $\mathcal{C}$, the scheduling algorithm in Section III, can stabilize the network independent of transport-layer ingress queue-based congestion control mechanism (as long as the minimum window size is one and the window sizes are bounded) and the (non-idling) service discipline used to transmit packets from active nodes.*

**Remark 1.** *Theorem 1 holds even when $h \equiv 1$ in (1), however, for the distributed implementation of the algorithm in Section V, we need $g$ to grow slightly slower than* $\log$.

Theorem 1 shows that it is possible to design the ingress queue-based congestion controller regardless of the scheduling algorithm implemented in the core network. This will allow using different congestion control mechanisms at the edge of the network for different fairness or QoS considerations without need to change the scheduling algorithm implemented at internal routers of the network. As we will see, a key ingredient of such a decomposition result is the use of difference between the logarithms of queue lengths, as in (2), for the link weights in the scheduling algorithm.

The rest of this section is devoted to proof of Theorem 1.

### A. Proof of Theorem 1

*1) Order of events:* Since we use a discrete-time model, we have to specify the order in which files/packets arrive and depart, which we do below:

1) At the beginning of each time slot, a scheduling decision is made by the scheduling algorithm. Packets depart from the MAC layers of scheduled links.
2) File arrivals occur next. Once a file arrives, a new TCP connection is set up for that file with an initial predetermined congestion window size.
3) For each TCP connection, if the congestion window is not full, packets are injected into the MAC layer from the Transport layer until the window size is fully used or there are no more packets at the Transport layer.

We re-index the files at the beginning of each time slot because some files might have been departed during the last time slot.

*2) State of the system:* Define the state of node $n$ as

$$\mathcal{S}_n(t) = \Big\{ (q_n^{(d)}(t), \mathcal{I}_n^{(d)}(t)) : d \in \mathcal{D}, (\xi_{nf}(t), \mathcal{W}_{nf}(t), \sigma_{nf}(t)) : 1 \leq f \leq N_n(t) \Big\}, \quad (7)$$

where $N_n(t)$ is the number of existing files at node $n$ at the beginning of time slot $t$, $\sigma_{nf}(t) \in \{1/\eta_1, \cdots, 1/\eta_K\}$ is its mean size (or type), and $\mathcal{W}_{nf}(t)$ is its corresponding congestion window size. Note that $\sigma_{nf}(t)$ is a function of time only because of re-indexing since a file might change its index from slot to slot. $\xi_{nf}(t) \in \{0,1\}$ indicates whether file $f$ has still packets in the Transport layer. More accurately, if $U_{nf}(t)$ is the number of remaining packets of file $f$ at node $n$, then $\xi_{nf}(t) = \mathbb{1}\{U_{nf}(t) > \mathcal{W}_{nf}(t)\}$. Obviously, if $n$ is not a source node, then we can remove $(\xi_{nf}, \mathcal{W}_{nf}, \sigma_{nf})$ from the description of $\mathcal{S}_n$. $\mathcal{I}_n^{(d)}(t)$ denotes the information required about $q_n^{(d)}(t)$ to serve the MAC-layer packets which depends on the specific service discipline implemented in MAC-layer queues. In the rest of the paper, we consider the case of FIFO (*First In-First Out*) service discipline in MAC-layer queues. In this case, $\mathcal{I}_n^{(d)}(t)$ is simply the ordering of packets in $q_n^{(d)}(t)$ according to their entrance times. As it will turn out from the proof, the system stability will hold for any none-idling service discipline. Define the state of the system to be $\mathcal{S}(t) = \{\mathcal{S}_n(t) : n \in \mathcal{N}\}$. Now, given the scheduling algorithm in section III, and our system model in Section II, $\mathcal{S}(t)$ evolves as a discrete-time Markov chain.

**Remark 2.** *We only require that the congestion window dynamics could be described as a function of queue lengths of the network so that the network Markov chain is well-defined. Even in the case that the congestion window is a function of the delayed queue lengths of the network up to T time slots earlier, due to the feedback delay of at most T from destination to source, the network state could be modified, to include the queues up to T time slots before, so that the same proof technique still applies.*

Next, we analyze the *Lyapunov drift* to show that the network Markov chain is positive recurrent and, as a result, the number of files in the system and queue sizes are stable.

*3) Lyapunov analysis:* Define $\bar{Q}_n^{(d)}(t) := \mathbb{E}\left[Q_n^{(d)}(t) | \mathcal{S}_n(t)\right]$ to be the expected total queue length at node $n$ given the state $\mathcal{S}_n(t)$. Then, if $n$ is a source, and $d$

is its destination,

$$\bar{Q}_n^{(d)}(t) = q_n^{(d)}(t) + \sum_{f=1}^{N_n(t)} \Big[\sigma_{nf}(t)\xi_{nf}(t)\Big]. \tag{8}$$

Otherwise, if $d \neq d(n)$ or $n$ is not a source, then $\bar{Q}_n^{(d)}(t) = q_n^{(d)}(t)$. Note that given the state $\mathcal{S}(t)$, $\bar{Q}_n^{(d)}$ is known.

The dynamics of $\bar{Q}_n^{(d)}(t)$ involves the dynamics of $q_n^{(d)}(t)$, $\xi_n(t)$, and $N_n(t)$, and, thus, it consists of: (i) departure of MAC-layer packets, (ii) new file arrivals (if $n$ is a source), (iii) arrival of packets from previous hops that use $n$ as an intermediate relay, (iv) injection of packets into the MAC layer (if $n$ is a source), and (v) departure of files from the Transport layer (if $n$ is a source). Hence,

$$\begin{aligned}\bar{Q}_n^{(d)}(t+1) &= \bar{Q}_n^{(d)}(t) - \sum_{j=1}^{N} R_{nj}^{(d)} z_{nj}^{(d)}(t) + \bar{A}_n^{(d)}(t) \\ &+ \sum_{i=1}^{N} R_{in}^{(d)} z_{in}^{(d)}(t) + \hat{A}_n^{(d)}(t) - \hat{D}_n^{(d)}(t), \quad (9)\end{aligned}$$

where $\bar{A}_n^{(d)}(t) = \sum_{f=N_n(t)+1}^{N_n(t)+a_n(t)} \sigma_{nf}(t)$ is the expected number of packet arrivals due to new files, $\hat{A}_n^{(d)}(t)$ is the total number of packets injected into the MAC layer to fill up the congestion window after scheduling and new file arrivals, and $\hat{D}_n^{(d)}(t) = \sum_{f=1}^{N_n(t)+a_n(t)} \sigma_{nf}(t) I_{nf}(t)$ is the Transport-layer "expected packet departure" because of the MAC-layer injections. Here, $I_{nf}(t) = 1$ indicates that the last packet of file $f$ *leaves the Transport layer during time slot* $t$; otherwise, $I_{nf}(t) = 0$[2]. Note that $\mathbb{E}\Big[\bar{A}_n^{(d)}(t)\Big] = \rho_n^{(d)}$.

Let $B_n^{(d)}(t) := \hat{A}_n^{(d)}(t) - \hat{D}_n^{(d)}(t)$ in (9), and $\mathbb{E}_{\mathcal{S}}[\cdot] := \mathbb{E}_{\mathcal{S}}[\cdot|\mathcal{S}(t)]$. It should be clear that when $n$ is a source but $d \neq d(n)$, or when $n$ is not a source, $\bar{A}_n^{(d)}(t) = \hat{A}_n^{(d)}(t) = \hat{D}_n^{(d)}(t) = B_n^{(d)}(t) \equiv 0$. Let $r_{max}$ denote the maximum link capacity over all the links in the network. Lemma 1 characterizes the first and second moments of $B_n^{(d)}(t)$.

**Lemma 1.** *For the process* $\{B_n^{(d)}(t)\}$,

(i) $\mathbb{E}_{\mathcal{S}(t)}\Big[B_n^{(d)}(t)\Big] = 0.$

(ii) *Let* $\eta_{min} = \min_{1\le i\le K} \eta_i$, *then*

$$\mathbb{E}_{\mathcal{S}(t)}\Big[B_n^{(d)}(t)^2\Big] \le (\kappa_n + N^2 r_{max}^2)\max\{\mathcal{W}_{cong}^2, 1/\eta_{min}^2\}.$$

Therefore, we can write

$$\begin{aligned}\bar{Q}_n^{(d)}(t+1) &= \bar{Q}_n^{(d)}(t) - \sum_{j=1}^{N} R_{nj}^{(d)} z_{nj}^{(d)}(t) + \tilde{A}_n^{(d)}(t) \\ &\quad + \sum_{i=1}^{N} R_{in}^{(d)} z_{in}^{(d)}(t), \quad (10)\end{aligned}$$

where $\tilde{A}_n^{(d)}(t) := \bar{A}_n^{(d)}(t) + B_n^{(d)}(t)$. Note that $\tilde{A}_n^{(d)}(t)$ has mean $\rho_n^{(d)}$ and finite second moment.

[2]To notice the difference between the indicators $I_{nf}(t)$ and $\xi_{nf}(t)$, consider a specific file and assume that its last packet enters the Transport layer at time slot $t_0$, departs the Transport layer during time slot $t_1$ and departs the MAC layer during time slot $t_2$, then its corresponding indicator $I$ is 1 at time $t_1$ and is 0 for $t_0 \le t < t_1$ and $t_1 < t \le t_2$, while its indicator $\xi$ is 0 for all time $t_1 \le t \le t_2$, and 1 for $t_0 \le t < t_1$.

Let $G(u) := \int_0^u g(x)dx$ for the function $g$ defined in (1). Then $G$ is a strictly convex and increasing function. Consider a Lyapunov function

$$V(\mathcal{S}(t)) = \sum_{n=1}^{N}\sum_{d\in\mathcal{D}} G(\bar{Q}_n^{(d)}(t)).$$

Let $\Delta V(t) := V(\mathcal{S}(t+1)) - V(\mathcal{S}(t))$. Using convexity and monotonicity of $G$, we get

$$\Delta V(t) \le \sum_{n=1}^{N}\sum_{d\in\mathcal{D}} g(\bar{Q}_n^{(d)}(t+1))\left(\bar{Q}_n^{(d)}(t+1) - \bar{Q}_n^{(d)}(t)\right).$$

Next, observe that, using (10),

$$|\bar{Q}_n^{(d)}(t+1) - \bar{Q}_n^{(d)}| \le \tilde{A}_n^{(d)}(t) + N r_{max}.$$

Hence, because $g$ is strictly increasing,

$$\begin{aligned}g(\bar{Q}_n^{(d)}(t+1)) &\le g\left(\bar{Q}_n^{(d)}(t) + \tilde{A}_n^{(d)}(t) + N r_{max}\right) \\ &\le g(\bar{Q}_n^{(d)}(t)) + (\tilde{A}_n^{(d)}(t) + N r_{max})\end{aligned}$$

where the last inequality follows from concavity of $g$ and the fact that $g' \le 1$. Hence,

$$\begin{aligned}\Delta V(t) &\le \sum_{n=1}^{N}\sum_{d\in\mathcal{D}} g(\bar{Q}_n^{(d)}(t))(\bar{Q}_n^{(d)}(t+1) - \bar{Q}_n^{(d)}(t)) \\ &+ \sum_{n=1}^{N}\sum_{d\in\mathcal{D}} (\tilde{A}_n^{(d)}(t) + N r_{max})^2.\end{aligned}$$

Define $u_n^{(d)}(t) := \max\left\{\sum_{j=1}^{N} R_{nj}^{(d)} x_{nj}^{(d)}(t) - q_n^{(d)}(t), 0\right\}$, to be the wasted service for packets of destination $d$, i.e., when $n$ is included in the schedule but it does not have enough packets of destination $d$ to transmit. Then, we have

$$\begin{aligned}\Delta V(t) &\le \sum_{n=1}^{N}\sum_{d\in\mathcal{D}} \Big\{ g(\bar{Q}_n^{(d)}(t))\Big[\sum_{i=1}^{N} R_{in}^{(d)} x_{in}^{(d)}(t) + \tilde{A}_n^{(d)}(t) \\ &- \sum_{j=1}^{N} R_{nj}^{(d)} x_{nj}^{(d)}(t)\Big]\Big\} + \sum_{n=1}^{N}\sum_{d\in\mathcal{D}} g(\bar{Q}_n^{(d)}(t)) u_n^{(d)}(t) \\ &+ \sum_{n=1}^{N}\sum_{d\in\mathcal{D}} (\tilde{A}_n^{(d)}(t) + N r_{max})^2.\end{aligned}$$

Taking the expectation of both sides, given the state at time $t$ is known, yields

$$\begin{aligned}\mathbb{E}_{\mathcal{S}(t)}\Big[\Delta V(t)\Big] &\le \sum_{n=1}^{N}\sum_{d\in\mathcal{D}} \Big\{ g(\bar{Q}_n^{(d)}(t))\mathbb{E}_{\mathcal{S}(t)}[\rho_n^{(d)} \\ &+ \sum_{i=1}^{N} R_{in}^{(d)} x_{in}^{(d)}(t) - \sum_{j=1}^{N} R_{nj}^{(d)} x_{nj}^{(d)}(t)]\Big\} \\ &+ \mathbb{E}_{\mathcal{S}(t)}\Big[\sum_{n=1}^{N}\sum_{d\in\mathcal{D}} g(\bar{Q}_n^{(d)}(t)) u_n^{(d)}(t)\Big] + C_1,\end{aligned}$$

where $C_1 = \mathbb{E}\Big[\sum_{n=1}^{N}\sum_{d\in\mathcal{D}} (\tilde{A}_n^{(d)}(t) + N r_{max})^2\Big] < \infty$, because $\mathbb{E}\Big[\tilde{A}_n^{(d)}(t)^2\Big] < \infty$.

**Lemma 2.** *There exists a positive constant $C_2$ such that, for all $\mathcal{S}(t)$, $\sum_{n=1}^{N}\sum_{d\in\mathcal{D}}\mathbb{E}_{\mathcal{S}(t)}\Big[g(\bar{Q}_n^{(d)}(t))u_n^{(d)}(t)\Big] \le C_2$.*

Using Lemma 2 and changing the order of summations, we have

$$\mathbb{E}_{\mathcal{S}(t)}\Big[\Delta V(t)\Big] \le C_1 + C_2 + \sum_{n=1}^{N}\sum_{d\in\mathcal{D}} g(\bar{Q}_n^{(d)}(t))\rho_n^{(d)}$$
$$-\mathbb{E}_{\mathcal{S}(t)}\Big[\sum_{(i,j)\in\mathcal{L}}\sum_{d\in\mathcal{D}} x_{ij}^{(d)}(t)(g(\bar{Q}_i^{(d)}(t)) - g(\bar{Q}_j^{(d)}(t)))\Big]. \quad (11)$$

Recall that the link weight that is actually used in the algorithm is based on the MAC-layer queues as in (2)-(3). For the analysis, we also define a new link weight based on the state as

$$W_{ij}(t) = \max_{d\in\mathcal{D}:R_{ij}^{(d)}=1} W_{ij}^{(d)}(t), \quad (12)$$

where, for a link $(i,j)\in\mathcal{L}$ with $R_{ij}^{(d)}=1$,

$$W_{ij}^{(d)}(t) := g(\bar{Q}_i^d(t)) - g(\bar{Q}_j^d(t)). \quad (13)$$

Then, the two types of link weights only differ by a constant as stated by the following lemma.

**Lemma 3.** *Let $W_{ij}(t)$ and $w_{ij}(t)$, $(i,j)\in\mathcal{L}$, be the link weights defined by (12)-(13) and (2)-(3) respectively. Then at all times*

$$|W_{ij}(t) - w_{ij}(t)| \le \frac{\log(1+1/\eta_{min})}{h(0)}.$$

*Proof:* Recall that, at each node $n$, for all destinations $d \neq d(n)$, we have $\bar{Q}_n^d(t) = q_n^d(t)$. If $d = d(n)$ is the destination of $n$, then $\bar{Q}_n^d(t)$ consists of: (i) packets of $d$ received from upstream flows that use $n$ as an intermediate relay, and (ii) MAC-layer packets received from the files generated at $n$ itself. Since $1 \le \mathcal{W}_{nf}(t) \le \mathcal{W}_{cong}$, the number of files with destination $d$ that are generated at node $n$ or have packets at node $n$ as an intermediate relay, is at most $q_n^{(d)}(t)$. Therefore, it is clear that $q_n^d(t) \le \bar{Q}_n^d(t) \le q_n^d(t) + q_n^d(t)\frac{1}{\eta_{min}}$. In the rest of the proof, we drop the dependence of queues on $t$ for compactness. For all $n$ and $d$, using a log-type function, as the function $g$ in (1), yields

$$\begin{aligned} g(q_n^d) \le g(\bar{Q}_n^d) &\le g\left(q_n^d(1+1/\eta_{min})\right) \\ &\le \frac{\log\left((1+q_n^d)(1+1/\eta_{min})\right)}{h(q_n^d(1+1/\eta_{min}))} \\ &\le g(q_n^d) + \frac{\log(1+1/\eta_{min})}{h(0)}. \end{aligned} \quad (14)$$

It then follows that, $\forall d\in\mathcal{D}$, and $\forall(i,j)\in\mathcal{L}$ with $R_{ij}^{(d)}=1$,

$$|W_{ij}^{(d)} - w_{ij}^{(d)}| \le \log(1+1/\eta_{min})/h(0). \quad (15)$$

Let $d_{ij}^* := \arg\max_{d:R_{ij}^{(d)}=1} W_{ij}^{(d)}$ and $\tilde{d}_{ij}^*$ as in (4). Then, using (15)

$$w_{ij} \ge w_{ij}^{(d_{ij}^*)} \ge W_{ij} - \log(1+1/\eta_{min})/h(0),$$

and, similarly,

$$W_{ij} \ge W_{ij}^{(\tilde{d}_{ij}^*)} \ge w_{ij} - \log(1+1/\eta_{min})/h(0).$$

This concludes the proof. ∎

Let $x^*(t)$ be the max weight schedule based on weights $\{W_{ij}(t):(i,j)\in\mathcal{L}\}$, i.e.,

$$x^*(t) = \arg\max_{x\in\mathcal{R}} \sum_{(i,j)\in\mathcal{L}} x_{ij}W_{ij}(t). \quad (16)$$

*Note the distinction between $x^*$ and $\tilde{x}^*$ as we used $\tilde{x}^*(t)$ in (5) to denote the max weight schedule based on MAC-layer queues.* The weights of the schedules $\tilde{x}^*$ and $x^*$ differ only by a constant for all queue values as we show next. From definition of $x^*$, in (16),

$$\sum_{(i,j)\in\mathcal{L}} x_{ij}^*W_{ij}(t) - \sum_{(i,j)\in\mathcal{L}} \tilde{x}_{ij}^*W_{ij}(t) \ge 0. \quad (17)$$

On the other hand,

$$\sum_{(i,j)\in\mathcal{L}} x_{ij}^*W_{ij}(t) - \sum_{(i,j)\in\mathcal{L}} \tilde{x}_{ij}^*W_{ij}(t) =$$
$$\sum_{(i,j)\in\mathcal{L}} x_{ij}^*W_{ij}(t) - \sum_{(i,j)\in\mathcal{L}} x_{ij}^*w_{ij}(t) \quad (18)$$
$$+\sum_{(i,j)\in\mathcal{L}} x_{ij}^*w_{ij}(t) - \sum_{(i,j)\in\mathcal{L}} \tilde{x}_{ij}^*w_{ij}(t) \quad (19)$$
$$+\sum_{(i,j)\in\mathcal{L}} \tilde{x}_{ij}^*w_{ij}(t) - \sum_{(i,j)\in\mathcal{L}} \tilde{x}_{ij}^*W_{ij}(t) \quad (20)$$
$$\le 2N^2 r_{max}\log(1+1/\eta_{min})/h(0), \quad (21)$$

because, by Lemma 3, (18) and (20) are less than $N^2 r_{max}\log(1+1/\eta_{min})/h(0)$ each, and (19) is negative by definition of $\tilde{x}^*$ in (5). Hence, using (11), (12), and (21), *under MAC scheduling $\tilde{x}^*$*, the Lyapunov drift is bounded as follows

$$\begin{aligned} \mathbb{E}_{\mathcal{S}(t)}\Big[\Delta V(t)\Big] &\le \sum_{n=1}^{N}\sum_{d\in\mathcal{D}}\Big\{g(\bar{Q}_n^{(d)}(t))\rho_n^{(d)}\Big\} \\ &- \mathbb{E}_{\mathcal{S}(t)}\Big[\sum_{(i,j)\in\mathcal{L}} x_{ij}^*(t)W_{ij}\Big] + C, \end{aligned}$$

where $C = C_1 + C_2 + 2N^2 r_{max}\log(1+1/\eta_{min})/h(0)$.

Accordingly, using (12)-(13), and changing the order of summations in the right hand side of the above inequality yields

$$\mathbb{E}_{\mathcal{S}(t)}\Big[\Delta V(t)\Big] \le$$
$$\sum_{n=1}^{N}\sum_{d\in\mathcal{D}}\Bigg\{g(\bar{Q}_n^{(d)}(t))\mathbb{E}_{\mathcal{S}(t)}\big[\rho_n^{(d)} + \sum_{i=1}^{N} R_{in}^{(d)}{x^*}_{in}^{(d)}(t)$$
$$-\sum_{j=1}^{N} R_{nj}^{(d)}{x^*}_{nj}^{(d)}(t)\big]\Bigg\} + C,$$

where ${x^*}_{ij}^{(d)}(t) = x_{ij}^*(t)$ for $d = d_{ij}^*(t)$ (ties are broken at random) and is zero otherwise. The rest of the proof is standard. Since load $\boldsymbol{\rho}$ is strictly inside the capacity region, there must exist a $\epsilon > 0$ and a $\gamma\in\mathrm{Co}(\mathcal{R})$ such that

$$\rho_n^{(d)} + \epsilon \le \sum_{j=1}^{N} R_{nj}^{(d)}\gamma_{nj}^{(d)} - \sum_{i=1}^{N} R_{in}^{(d)}\gamma_{in}^{(d)} \;;\forall n\in\mathcal{N}, \forall d\in\mathcal{D}. \quad (22)$$

Hence, for any $\delta > 0$,

$$\mathbb{E}_{\mathcal{S}(t)}\Big[\Delta V(t)\Big] \leq$$
$$\sum_{n=1}^{N}\sum_{d\in\mathcal{D}} g(\bar{Q}_n^{(d)}(t))\left[\sum_{i=1}^{N} R_{in}^{(d)} {x^*}_{in}^{(d)}(t) - \sum_{j=1}^{N} R_{nj}^{(d)} {x^*}_{nj}^{(d)}(t)\right]$$
$$- \sum_{n=1}^{N}\sum_{d\in\mathcal{D}} g(\bar{Q}_n^{(d)}(t))\left[\sum_{i=1}^{N} R_{in}^{(d)} \gamma_{in}^{(d)}(t) - \sum_{j=1}^{N} R_{in}^{(d)} \gamma_{nj}^{(d)}(t)\right]$$
$$- \epsilon\sum_{n=1}^{N}\sum_{d\in\mathcal{D}} g(\bar{Q}_n^{(d)}(t)) + C.$$

But from definition of $x^*(t)$ and convexity of $\mathrm{Co}(\mathcal{R})$, $\sum_{(i,j)\in\mathcal{L}} x^*_{ij} W_{ij}(t) \geq \sum_{(i,j)\in\mathcal{L}} \gamma_{ij} W_{ij}(t),\ \forall \gamma \in \mathrm{Co}(\mathcal{R})$, hence,

$$\mathbb{E}_{\mathcal{S}(t)}\Big[\Delta V(t)\Big] \leq -\epsilon\sum_{n=1}^{N}\sum_{d\in\mathcal{D}} g(\bar{Q}_n^{(d)}(t)) + C \leq -\delta,$$

whenever $\max_{n,d} \bar{Q}_n^{(d)} \geq g^{-1}\big(\frac{C_2+\delta}{\epsilon}\big)$ or, as a sufficient condition, whenever $\max_{n,d} q_n^{(d)} \geq g^{-1}\big(\frac{C_2+\delta}{\epsilon}\big)$. Therefore, it follows that the system is stable by an extension of the Foster-Lyapunov criteria [16] (Theorem 3.1 in [1]). In particular, queue sizes and the number of files in the system are stable.

**Remark 3.** *Although we have assumed that file sizes follow a mixture of geometric distributions, our results also hold for the case of bounded file sizes with general distribution. The proof argument for the latter case is obtained by minor modifications of the proof presented in this paper (see [23]) and, hence, has been omitted for brevity.*

## V. Distributed Implementation

The optimal scheduling algorithm in Section III requires us to find a maximum weight-type schedule at each time, i.e., we need to solve (5) at every time. This is a formidable task, hence, in this section, we design a distributed version of the algorithm based on *Glauber dynamics*.

For simplicity, we consider the following criterion for successful packet reception: packet transmission over link $(i,j) \in \mathcal{L}$ is successful if none of the neighbors of node $j$ are transmitting. Furthermore, we assume that every node can transmit to at most one node at each time, receive from at most one node at each time, and cannot transmit and receive simultaneously (over the same frequency band). This especially models the packet reception in the case that the set of neighbors of node $i$, i.e., $C(i) = \{j : (i,j) \in \mathcal{L}\}$, is the set of nodes that are within the transmission range of $i$ and the interference caused by $i$ at all other nodes, except its neighbors, is negligible. Moreover, the packet transmission over $(i,j)$ is usually followed by an ACK transmission from receiver to sender, over $(j,i)$. Hence, for a *synchronized data/ACK system*, we can define a *Conflict Set* (CS) for link $(i,j)$ as

$$\begin{aligned}\mathrm{CS}_{(i,j)} \quad = \quad & \{(a,b) \in \mathcal{L} : a \in C(j), \text{ or } b \in C(i), \\ & \text{or } a \in \{i,j\}, \text{ or } b \in \{i,j\}\}. \end{aligned} \tag{23}$$

This ensures that when the links in $\mathrm{CS}_{(i,j)}$ are inactive, the data/ACK transmission over $(i,j)$/$(j,i)$ is successful.

Furthermore, for simplicity, assume that in each time slot, at most one packet could be successfully transmitted over a link $(i,j)$, i.e., $x_{ij}(t) \in \{0,1\}$. We can represent the interference constraints by using a *conflict graph* $\mathcal{G}(\mathcal{V},\mathcal{E})$, where each vertex in $\mathcal{V}$ is a communication link in the wireless network. There is an edge $((i,j),(a,b)) \in \mathcal{E}$ between vertices $(i,j)$ and $(a,b)$ if simultaneous transmissions over communication links $(i,j)$ and $(a,b)$ are not successful. Therefore, at each time slot, the active links should form an *independent set* of $\mathcal{G}$, i.e., no two scheduled vertices can share an edge in $\mathcal{G}$. Let $\mathcal{R}$ be the set of all such feasible schedules and $|\mathcal{L}|$ denote the number of communication links in the wireless network.

We say that a node is active if it is a sender or a receiver for some active link. Inactive nodes can *sense the wireless medium* and know if there is an active node in their neighborhood. This is possible because we use a synchronized data/ACK system and detecting active nodes can be performed by sensing the data transmission of active senders and sensing the ACK transmission of active receivers. Hence, using such carrier sensing, nodes $i$ and $j$ know if the channel is idle, i.e., $\sum_{(a,b)\in\mathrm{CS}_{(i,j)}} x_{ab}(t) = 0$, or if the channel is busy, i.e., $\sum_{(a,b)\in\mathrm{CS}_{(i,j)}} x_{ab}(t) \geq 1$.

**Remark 4.** *For the case of single hop networks, the link weight (3) is reduced to* $w_{ij}(t) = g(1+q_i(t))/h(q_i(t))$ *where* $i$ *is the source and* $j$ *is the destination of flow over* $(i,j)$*. Such a weight function is exactly the one that under which throughput optimality of CSMA has been established in [10]. Next, we will propose a version of CSMA that is suitable for the general case of multihop flows and will prove its throughput optimality. The proof uses techniques originally developed in [12], [13] for continuous-time CSMA algorithms, and adapted in [10] for the discrete-time model considered here.*

### A. Basic CSMA Algorithm for Multihop Networks

For our algorithm, *based on the MAC layer information*, we define a modified weight for each link $(i,j)$ as

$$\tilde{w}_{ij}(t) = \max_{d: R_{ij}^{(d)}=1} \tilde{w}_{ij}^{(d)}(t), \tag{24}$$

where

$$\tilde{w}_{ij}^{(d)}(t) = \tilde{g}\left(q_i^{(d)}(t)\right) - \tilde{g}\left(q_j^{(d)}(t)\right), \tag{25}$$

and,

$$\tilde{g}\left(q_i^{(d)}(t)\right) = \max\left\{g\left(q_i^{(d)}(t)\right), g^*(t)\right\}, \tag{26}$$

where the function $g$ is the same as (1) defined for the centralized algorithm, and

$$g^*(t) := \frac{\epsilon}{4N^3} g(q_{max}(t)), \tag{27}$$

where $q_{max}(t) := \max_{i,d} q_i^{(d)}(t)$ is the maximum MAC-layer queue length in the network at time $t$ and assumed to be known, and $\epsilon$ is an arbitrary small but fixed positive number. Note that if we remove $g^*(t)$ from the above definition, then $\tilde{w}_{ij}$ is equal to $w_{ij}$ in (2)-(3).

Consider the conflict graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ of the network as defined earlier. At each time slot $t$, a link $(i,j)$ is chosen uniformly at random, then

(i) If $\tilde{x}_{ab}(t-1) = 0$ for all links $(a,b) \in \mathrm{CS}_{(i,j)}$, then $\tilde{x}_{ij}(t) = 1$ with probability $p_{ij}(t)$, and $\tilde{x}_{ij}(t) = 0$ with probability $1 - p_{ij}(t)$ .
Otherwise, $\tilde{x}_{ij}(t)$=0.

(ii) $\tilde{x}_{ab}(t) = x_{ab}(t-1)$ for all $(a,b) \neq (i,j)$.

(iii) $x_{ij}^{(d)}(t) = \tilde{x}_{ij}(t)$ if $d = \arg\max_{d:R_{ij}^{(d)}=1} \tilde{w}_{ij}^{(d)}(t)$ (break ties at random), and zero otherwise.

We choose $p_{ij}(t)$ to be

$$p_{ij}(t) = \frac{\exp(\widetilde{w}_{ij}(t))}{1 + \exp(\widetilde{w}_{ij}(t))}. \tag{28}$$

It turns out that the choice of function $g$ is crucial in establishing the throughput optimality of the algorithm for general networks. The following theorem states the main result regarding the throughput optimality of the basic CSMA algorithm.

**Theorem 2.** *Under the function $g$ specified in (1), the basic CSMA algorithm, with any $\epsilon > 0$, can stabilize the network for any $\boldsymbol{\rho} \in (1-3\epsilon)\mathcal{C}$, independent of Transport-layer ingress queue-based congestion control (as long as the minimum window size is one and the window sizes are bounded) and the (non-idling) service discipline used to serve packets of active queues.*

### *B. Distributed Implementation*

The basic algorithm is based on Glauber-Dynamics with one site update at each time. For distributed implementation, we need a randomized mechanism to select a link uniformly at random at each time slot. We use the Q-CSMA idea [11] to perform the link selection: each time slot is divided into a control slot and a data slot. In the control slot, nodes exchange short control messages, similar to RTS/CTS packets in IEEE 802.11 protocol, to come up with a collision-free *decision schedule* $m$. In the data slot, each link $(i,j)$ that is included in the decision schedule performs the basic CSMA algorithm. See [11], [25] for complete details.

The control message sent from node $j$ to $i$ in time slot $t$, contains the carrier sense information of node $j$ at time $t-1$, and the vector of MAC-layer queue sizes of node $j$ at time $t$, i.e., $[q_j^{(d)}(t) : d \in \mathcal{D}]$, to determine the weight of link $(i,j)$.

**Remark 5.** *To determine the weight at each link, $q_{max}(t)$ is also needed. Instead, each node can maintain an estimate of $q_{max}(t)$ similar to the procedure suggested in [12]. In fact, it is easy to incorporate such a procedure in our algorithm because, in the control slot, each node can include its estimate of $q_{max}(t)$ in the control messages and update its estimate based on the received control messages. Then we can use Lemma 2 of [12] to complete the stability proof. So we do not pursue this issue here. In practical networks $\frac{\epsilon}{4N^3}\log(1+q_{max}(t))$ is small and we can use the weight function $g$ directly, and thus, there may not be any need to know $q_{max}(t)$.*

**Corollary 1.** *Under the weight function $g$ specified in (1), the distributed algorithm can stabilize the network for any $\boldsymbol{\rho} \in (1-3\epsilon)\mathcal{C}$.*

The rest of this section is devoted to proof of Theorem 2. The proof of Corollary 1 is almost identical and omitted for brevity (see [10] and [25] for all the details).

### *C. Proof of Theorem 2*

First, suppose the weights are constants, i.e., the basic algorithm uses a weight vector $\tilde{\mathbf{w}} = [\tilde{w}_{ij} : (i,j) \in \mathcal{L}]$ at all times. Then, the basic algorithm is essentially an irreducible, aperiodic, and reversible Markov chain (called Glauber dynamics) to generate the independent sets of $\mathcal{G}(\mathcal{V}, \mathcal{E})$. So, the state space $\mathcal{R}$ consists of all independent sets of $\mathcal{G}$. The stationary distribution of the chain is given by

$$\pi(s) = \frac{1}{Z} \exp\Big( \sum_{(i,j)\in s} \tilde{w}_{ij} \Big); \quad s \in \mathcal{R}, \tag{29}$$

where $Z$ is the normalizing constant.

We start with the following lemma that relates the modified link weight and the original link weight.

**Lemma 4.** *For all links $(i,j) \in \mathcal{L}$, the link weights (24) and (3) differ at most by $g^*(t)$, i.e.,*

$$|\tilde{w}_{ij}(t) - w_{ij}(t)| \leq g^*(t). \tag{30}$$

Proof is simple and has been omitted for brevity (see [25]). The basic algorithm uses a time-varying version of the Glauber dynamics, where the weights change with time. This yields a time-inhomogeneous Markov chain but we will prove that, for the choice of function $g$ in (1), it behaves similarly to the Glauber dynamics. The proof consists of 4 steps; steps 1-3 are concentred with properties of the basic CSMA algorithm and step 4 is the Lyapunov analysis.

*1) Mixing time of Glauber dynamics:* The eigenvalues of the corresponding transition probability matrix $\mathbf{P}$ can be ordered in such a way that

$$\lambda_1 = 1 > \lambda_2 \geq ... \geq \lambda_{|\mathcal{R}|} > -1.$$

The convergence to steady state distribution is geometric with a rate equal to the *Second Largest Eigenvalue Modulus* (SLEM) of $\mathbf{P}$ [14]. In fact, for any initial probability distribution $\mu_0$ on $\mathcal{R}$, and for all $t \geq 1$,

$$\|\mu_0 \mathbf{P}^t - \pi\|_{\frac{1}{\pi}} \leq (\lambda^*)^t \|\mu_0 - \pi\|_{\frac{1}{\pi}}, \tag{31}$$

where $\lambda^* = \max\{\lambda_2, |\lambda_{|\mathcal{R}|}|\}$ is the SLEM. Note that, by definition, $\|z\|_{1/\pi} = \left(\sum_{i=1}^r z(i)^2 \frac{1}{\pi(i)}\right)^{1/2}$.

The following lemma gives an upper bound on the SLEM $\lambda^*$ of Glauber dynamics.

**Lemma 5.** *For the Glauber dynamics with the weight vector $\tilde{\mathbf{w}}$ on a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$,*

$$\lambda^* \leq 1 - \frac{1}{16^{|\mathcal{V}|} \exp(4|\mathcal{V}|\tilde{w}_{max})},$$

*where $\tilde{w}_{max} = \max_{(i,j)\in\mathcal{L}} \tilde{w}_{ij}$.*

See the appendix of [25] or [10] for the proof. We define the *mixing time* as $T = \frac{1}{1-\lambda^*}$, so

$$T \le 16^{|\mathcal{L}|} \exp(4|\mathcal{L}|\tilde{w}_{max}) \tag{32}$$

Simple calculation, based on (31), reveals that the amount of time needed to get close to the stationary distribution is *approximately* proportional to $T$.

*2) A key proposition:* At any time slot $t$, given the weight vector $\tilde{\mathbf{w}}(t) = [\tilde{w}_{ij}(t) : (i,j) \in \mathcal{L}]$, the centralized algorithm, described in Section III, should solve $\max_{s\in\mathcal{R}} \sum_{(i,j)\in s} \tilde{w}_{ij}(t)$, instead, the distributed algorithm tries to simulate a distribution

$$\pi_t(s) = \frac{1}{Z_t} \exp\Big(\sum_{(i,j)\in s} \tilde{w}_{ij}(t)\Big); \quad s \in \mathcal{R}, \tag{33}$$

i.e., the stationary distribution of Glauber dynamics with the weight vector $\tilde{\mathbf{w}}(t)$ at time $t$.

Let $\mathbf{P}_t$ denote the transition probability matrix of Glauber dynamics with the weight vector $\tilde{\mathbf{w}}(t)$. Also let $\mu_t$ be the true probability distribution of the time-inhomogeneous chain, over the set of schedules $\mathcal{R}$, at time $t$. Therefore, $\mu_t = \mu_{t-1}\mathbf{P}_t$. Let $\pi_t$ denote the stationary distribution of the time-homogenous Markov chain with $\mathbf{P} = \mathbf{P}_t$ as in (33). By choosing proper $g^*$ and $g(\cdot)$, we aim to ensure that $\mu_t$ and $\pi_t$ are close enough, i.e., $\|\pi_t - \mu_t\|_{TV} \le \delta$ for some $\delta$ arbitrary small, where $\|\pi - \mu\|_{TV} := \frac{1}{2}\sum_{i=1}^{r} |\pi(i) - \mu(i)|$. Note that $\|\mu - \pi\|_{\frac{1}{\pi}} \ge 2\|\mu - \pi\|_{TV}$. Next, we characterize the amount of change in the stationary distribution as a result of queue evolutions.

**Lemma 6.** *For any schedule $s \in \mathcal{R}$, $e^{-\alpha_t} \le \frac{\pi_{t+1}(s)}{\pi_t(s)} \le e^{\alpha_t}$, where,*

$$\alpha_t = 2(1+\mathcal{W}_{cong})|\mathcal{L}|g'\Big(g^{-1}(g^*(t+1)) - 1 - \mathcal{W}_{cong}\Big), \tag{34}$$

*and $\mathcal{W}_{cong}$ is the maximum congestion window size.*

Now, equipped with Lemmas 5 and 6, we make use of the results in [12], [13] and [10] in the final step of the proof. Specifically, we will use the following key Proposition from [10].

**Proposition 1.** *Given any $\delta > 0$, $\|\pi_t - \mu_t\|_{TV} \le \delta/4$ holds when $q_{max}(t) \ge q_{th} + t^*$, if there exists a $q_{th}$ such that*

$$\alpha_t T_{t+1} \le \delta/16 \text{ whenever } q_{max}(t) > q_{th}, \tag{35}$$

*where*

(i) $T_t \le 16^{|\mathcal{L}|} \exp(4|\mathcal{L}|\tilde{w}_{max}(t))$,
(ii) *$t^*$ is the smallest $t$ such that*

$$\frac{1}{\sqrt{\min_s \pi_{t_1}(s)}} \exp(-\sum_{k=t_1}^{t_1+t^*} \frac{1}{T_k^2}) \le \delta/4, \tag{36}$$

*with $q_{max}(t_1) = q_{th}$.*

In other words, Proposition 1 states that when queue lengths are large, the observed distribution of the schedules is close to the desired stationary distribution. The key idea in the proof is that, for $\alpha_t$ small, the weights change at the rate $\alpha_t$ while the system responds to these changes at the rate $1/T_{t+1}$. Condition (35) is to ensure that the weight dynamics are slow enough compared to response time of the chain such that the chain remains close to its equilibrium (stationary distribution).

We will also use the following lemma that relates the maximum queue length and the maximum weight in the network. Hence, when one grows, the other one increases as well.

**Lemma 7.** *Let $w_{max}(t) = \max_{(i,j)\in\mathcal{L}} w_{ij}(t)$. Then*

$$\frac{1}{N} g\left(q_{max}(t)\right) \le w_{max}(t) \le g\left(q_{max}(t)\right).$$

*3) Some useful properties of the basic CSMA algorithm:*

**Lemma 8.** *The Basic CSMA algorithm, with function $g$ as in (1), satisfies the requirements of Proposition 1.*

The formal proof can be found in the appendix. Roughly speaking, since the mixing time $T$ is exponential in $g(q_{max})$, $g'(g^{-1}(g^*))$ must be in the form of $e^{-g^*}$; otherwise it will be impossible to satisfy $\alpha_t T_{t+1} < \delta/16$ in Proposition 1 for any arbitrarily small $\delta$ as $q_{max}(t) \to \infty$. The only function with such a property is the $\log(\cdot)$ function. In fact, $g$ must grow slightly slower than $\log(\cdot)$ to satisfy (35), and to ensure the existence of a finite $t^*$ in Lemma 1. For example, by choosing functions that grow much slower than $\log(1+x)$, like $h(x) = \log(e + \log(1+x))$, we can make $g(x)$ behave approximately like $\log(1+x)$ for large ranges of $x$ (correspondingly, for the range of practical queue lengths).

Next, the following lemma states that, with high probability, the basic CSMA algorithm chooses schedules that their weights are close to the max weight schedule.

**Lemma 9.** *Given any $0 < \varepsilon < 1$ and $0 < \delta < 1$, there exists a $B(\delta, \varepsilon) > 0$ such that whenever $q_{max}(t) > B(\delta, \varepsilon)$, the basic CSMA algorithm chooses a schedule $s(t) \in \mathcal{R}$ such that*

$$\sum_{(i,j)\in s(t)} w_{ij}(t) \ge (1-\epsilon) \max_{s\in\mathcal{R}} \sum_{(i,j)\in s} w_{ij}(t),$$

*with probability larger than $1 - \delta$.*

*Proof:* Let $w^*(t) = \max_{s\in\mathcal{R}} \sum_{(i,j)\in s} w_{ij}(t)$ and define

$$\chi_t := \Big\{s \in \mathcal{R} : \sum_{(i,j)\in s} w_{ij}(t) < (1-\epsilon) w^*(t)\Big\}.$$

Therefore, we need to show that $\mu_t(\chi_t) \le \delta$, for $q_{max}(t)$ large enough. For our choice of $g(\cdot)$ and $g^*$, it follows from Proposition 1 that, whenever $q_{max}(t) > q_{th} + t^*$, $2\|\mu_t - \pi_t\|_{TV} \le \delta/2$, and consequently, $\sum_{s\in\mathcal{R}} \Big|\mu_t(s) - \pi_t(s)\Big| \le \delta/2$. Thus,

$$\sum_{s\in\chi_t} \mu_t(s) \le \sum_{s\in\chi_t} \pi_t(s) + \delta/2.$$

Therefore, to ensure that $\sum_{s\in\chi_t} \mu_t(s) \le \delta$, it suffices to have $\sum_{s\in\chi_t} \pi_t(s) \le \delta/2$. But, by Lemma 4, $\widetilde{w}_{ij}(t) \le w_{ij}(t) + g^*(t)$, so,

$$\begin{aligned}\sum_{s\in\chi_t} \pi_t(s) &\le \sum_{s\in\chi_t} \frac{1}{Z_t} e^{\sum_{(i,j)\in s} w_{ij}(t)} e^{|s| g^*(t)} \\ &\le \sum_{s\in\chi_t} \frac{1}{Z_t} e^{(1-\varepsilon) w^*(t)} e^{|\mathcal{L}| g^*(t)},\end{aligned}$$

and

$$Z_t = \sum_{s\in\mathcal{R}} e^{\sum_{(i,j)\in s}\tilde{w}_{ij}(t)} > \sum_{s\in\mathcal{R}} e^{\sum_{(i,j)\in s}(w_{ij}(t)-g^*(t))} > e^{w^*(t)-|\mathcal{L}|g^*(t)}.$$

Therefore,

$$\sum_{s\in\chi_t}\pi_t(s) \leq 2^{|\mathcal{L}|}e^{2|\mathcal{L}|g^*(t)-\varepsilon w^*(t)},$$

when $q_{max}(t) > q_{th} + t^*$. Note that $w^*(t) \geq w_{max}(t) \geq g(q_{max}(t))/N$, and $g^*(t) = \frac{\epsilon}{4N^3}g(q_{max}(t))$, so

$$\sum_{s\in\chi_t}\pi_t(s) \leq 2^{N^2}e^{-\frac{\epsilon}{2N}g(q_{max}(t))} \leq \delta/2$$

whenever $q_{max}(t) > B(\delta,\epsilon)$ with

$$B(\delta,\epsilon) = \max\Big\{q_{th}+t^*, g^{-1}\Big(\frac{2N}{\epsilon}(N^2\log 2+\log\frac{2}{\delta})\Big)\Big\}.$$

■

*4) Lyapunov analysis:* Now we are ready to prove the stability of the network under the basic CSMA algorithm. Let $x^*$ and $\tilde{x}^*$ be the optimal schedules based on total queues and MAC queues respectively, given by (16) and (5), and $\tilde{x}$ be the schedule generated by the basic CSMA algorithm. The proof is parallel to the stability argument of the centralized algorithm. In particular, the inequality (11) still holds, which is

$$\begin{aligned}\mathbb{E}_{\mathcal{S}(t)}\Big[\Delta V(t)\Big] &\leq C_1 + C_2 + \sum_{n=1}^{N}\sum_{d\in\mathcal{D}} g(\bar{Q}_n^{(d)}(t))\rho_n^{(d)} \\ &- \mathbb{E}_{\mathcal{S}(t)}\Big[\sum_{(i,j)\in\mathcal{L}}\sum_{d\in\mathcal{D}} x_{ij}^{(d)}(t)W_{ij}^{(d)}\Big] \\ &= C_1 + C_2 + \sum_{n=1}^{N}\sum_{d\in\mathcal{D}} g(\bar{Q}_n^{(d)}(t))\rho_n^{(d)} \\ &- \mathbb{E}_{\mathcal{S}(t)}\Big[\sum_{(i,j)\in\mathcal{L}} \tilde{x}_{ij}(t)W_{ij}(t)\Big]. \end{aligned} \tag{37}$$

Next, observe that

$$\sum_{(i,j)\in\mathcal{L}} x_{ij}^* W_{ij}(t) - \mathbb{E}_{\mathcal{S}(t)}\Big[\sum_{(i,j)\in\mathcal{L}}\tilde{x}_{ij}W_{ij}(t)\Big] =$$

$$= \mathbb{E}_{\mathcal{S}(t)}\Big[\sum_{(i,j)\in\mathcal{L}} x_{ij}^* W_{ij}(t) - \sum_{(i,j)\in\mathcal{L}} x_{ij}^* w_{ij}(t)\Big] \tag{38}$$

$$+ \mathbb{E}_{\mathcal{S}(t)}\Big[\sum_{(i,j)\in\mathcal{L}} x_{ij}^* w_{ij}(t) - \sum_{(i,j)\in\mathcal{L}} \tilde{x}_{ij} w_{ij}(t)\Big] \tag{39}$$

$$+ \mathbb{E}_{\mathcal{S}(t)}\Big[\sum_{(i,j)\in\mathcal{L}} \tilde{x}_{ij} w_{ij}(t) - \sum_{(i,j)\in\mathcal{L}} \tilde{x}_{ij} W_{ij}(t)\Big]. \tag{40}$$

Each of the terms (38) and (40) are less than $|\mathcal{L}|\log(1+1/\eta_{min})/h(0)$ by Lemma 3. The term (39) is bounded from above, by using Lemma 9, as follows.

$$\begin{aligned}(39) &\leq \sum_{(i,j)\in\mathcal{L}} x_{ij}^* w_{ij}(t) - (1-\delta)(1-\epsilon)\sum_{(i,j)\in\mathcal{L}} \tilde{x}_{ij}^* w_{ij}(t) \\ &\leq \sum_{(i,j)\in\mathcal{L}} x_{ij}^* w_{ij}(t) - (1-\delta)(1-\epsilon)\sum_{(i,j)\in\mathcal{L}} x_{ij}^* w_{ij}(t) \\ &\leq (1-(1-\delta)(1-\epsilon))\sum_{(i,j)\in\mathcal{L}} x_{ij}^* W_{ij}(t) \\ &\quad + |\mathcal{L}|\log(1+1/\eta_{min})/h(0),\end{aligned}$$

whenever $q_{max}(t) \geq B(\delta,\epsilon)$, for any $\delta > 0$. Thus, using the above bounds for terms (38), (39) and (40), we get

$$\begin{aligned}\mathbb{E}_{\mathcal{S}(t)}\Big[\sum_{(i,j)\in\mathcal{L}}\tilde{x}_{ij}W_{ij}(t)\Big] &\geq (1-\delta)(1-\epsilon)\sum_{(i,j)\in\mathcal{L}} x_{ij}^* W_{ij}(t) \\ &- 3|\mathcal{L}|\log(1+1/\eta_{min})/h(0).\end{aligned} \tag{41}$$

Using (41) in (37) yields

$$\begin{aligned}\mathbb{E}_{\mathcal{S}(t)}\Big[\Delta V(t)\Big] &\leq C_3 + \sum_{n=1}^{N}\sum_{d\in\mathcal{D}} g(\bar{Q}_n^{(d)}(t))\rho_n^{(d)} \\ &- (1-\delta)(1-\epsilon)\sum_{(i,j)\in\mathcal{L}} x_{ij}^* W_{ij}(t)\end{aligned} \tag{42}$$

where $C_3 := C_1 + C_2 + 3|\mathcal{L}|\log(1+1/\eta_{min})/h(0)$. Using (12) and rewriting the right-hand-side of (42), by changing the order of summations, yields

$$\mathbb{E}_{\mathcal{S}(t)}\Big[\Delta V(t)\Big] \leq C_3 + \sum_{n=1}^{N}\sum_{d\in\mathcal{D}} g(\bar{Q}_n^{(d)}(t))\Big[\rho_n^{(d)} + (1-\delta)(1-\epsilon)\Big(\sum_{i=1}^{N} R_{in}^{(d)} x^{*(d)}_{in}(t) - \sum_{j=1}^{N} R_{nj}^{(d)} x^{*(d)}_{nj}(t)\Big)\Big].$$

whenever $q_{max}(t) \geq B(\delta,\epsilon)$. The rest of the proof is standard. For any load $\boldsymbol{\rho}$ strictly inside $(1-3\epsilon)\mathcal{C}$, there must exist a $\gamma \in \text{Co}(\mathcal{R})$ such that for all $1 \leq n \leq N$, and all $d \in \mathcal{D}$,

$$\rho_n^{(d)} < (1-3\epsilon)\Big(\sum_{j=1}^{N} R_{nj}^{(d)}\gamma_{nj}^{(d)} - \sum_{i=1}^{N} R_{in}^{(d)}\gamma_{in}^{(d)}\Big). \tag{43}$$

Let $\frac{\rho^*}{1-3\epsilon} = \min_{n\in\mathcal{N},d\in\mathcal{D}}\Big(\sum_j R_{nj}^{(d)}\gamma_{nj}^{(d)} - \sum_i R_{in}^{(d)}\gamma_{in}^{(d)}\Big)$ for some positive $\rho^*$. Hence,

$$\begin{aligned}\mathbb{E}_{\mathcal{S}(t)}\Big[\Delta V(t)\Big] &\leq (1-\delta)(1-\epsilon)\sum_{n=1}^{N}\sum_{d\in\mathcal{D}}\Big\{g(\bar{Q}_n^{(d)}(t)) \times \\ &\quad \Big[\sum_{i=1}^{N} R_{in}^{(d)} x^{*(d)}_{in}(t) - \sum_{j=1}^{N} R_{nj}^{(d)} x^{*(d)}_{nj}(t)\Big]\Big\} \\ &+ (1-3\epsilon)\sum_{n=1}^{N}\sum_{d\in\mathcal{D}}\Big\{g(\bar{Q}_n^{(d)}(t)) \times \\ &\quad \Big[\sum_{j=1}^{N} R_{nj}^{(d)}\gamma_{nj}^{(d)} - \sum_{i=1}^{N} R_{in}^{(d)}\gamma_{in}^{(d)}\Big]\Big\} + C_3.\end{aligned}$$

For any fixed small $\epsilon > 0$, we can choose $\delta < \epsilon/(1-\epsilon)$ to ensure $(1-\delta)(1-\epsilon) > 1-2\epsilon$. Moreover, from definition of

$x^*(t)$ and convexity of $\mathrm{Co}(\mathcal{R})$, it follows that

$$\sum_{n=1}^{N}\sum_{d\in\mathcal{D}} g(\bar{Q}_n^{(d)}(t))\Big[\sum_{j=1}^{N} R_{nj}^{(d)} {x^*}_{nj}^{(d)}(t) - \sum_{i=1}^{N} R_{in}^{(d)} {x^*}_{in}^{(d)}(t)\Big]$$
$$\geq \sum_{n=1}^{N}\sum_{d\in\mathcal{D}} g(\bar{Q}_n^{(d)}(t))\Big[\sum_{j=1}^{N} R_{nj}^{(d)}\gamma_{nj}^{(d)} - \sum_{i=1}^{N} R_{in}^{(d)}\gamma_{in}^{(d)}\Big], \quad (44)$$

for any $\gamma \in \mathrm{Co}(\mathcal{R})$. Hence,

$$\mathbb{E}_{\mathcal{S}(t)}\Big[\Delta V(t)\Big] \leq$$
$$-\epsilon\sum_{n=1}^{N}\sum_{d\in\mathcal{D}} g(\bar{Q}_n^{(d)}(t))\Big[\sum_{j=1}^{N} R_{nj}^{(d)}\gamma_{nj}^{(d)} - \sum_{i=1}^{N} R_{in}^{(d)}\gamma_{in}^{(d)}\Big] + C_3$$
$$\leq -\rho^*\frac{\epsilon}{1-3\epsilon}\sum_{n=1}^{N}\sum_{d\in\mathcal{D}} g(\bar{Q}_n^{(d)}(t)) + C_3 \leq -\epsilon',$$

whenever $\max_{n,d}\bar{Q}_n^{(d)} \geq g^{-1}\big(\frac{C_3+\epsilon'}{\rho^*}\frac{1-3\epsilon}{\epsilon}\big)$ and $q_{max}(t) \geq B(\delta,\epsilon)$ or, as a sufficient condition, whenever

$$q_{max}(t) \geq \max\Big\{B(\delta,\epsilon), g^{-1}\big(\frac{C_3+\epsilon'}{\rho^*}\frac{1-3\epsilon}{\epsilon}\big)\Big\}.$$

In particular, to get negative drift, $-\epsilon'$, for some positive constant $\epsilon'$, it suffices that

$$\max_n N_n > \max\Big\{g^{-1}\big(\frac{C_3+\epsilon'}{\rho^*}\frac{1-3\epsilon}{\epsilon}\big), B(\delta,\epsilon)\Big\}$$

because $q_{max}(t) \geq \max_n N_n$, and $g$ is an increasing function. This concludes the proof of Theorem 2.

## VI. Concluding remarks

In this paper, we showed that $\alpha$-fair congestion control is not necessary for flow-level stability. In fact, by using back-pressure with link weights that are log-differentials of (MAC-layer) queue lengths, the network stability is guaranteed for very general congestion control mechanisms. Hence, one can use different congestion control mechanisms for providing different QoS, without need to change the scheduling algorithm implemented at the internal routers of the network. The choice of log-differential link weights also enables us to implement our algorithm in a distributed fashion using CSMA schemes, without loss of throughput optimality.

Our constraining assumptions regarding the congestion control mechanisms are very mild and compatible with the standard implementations like TCP. It is observed in [20] in the context of multiclass queueing systems that a fixed congestion window size implicitly solves an optimization problem in an asymptotic regime. It would be interesting to investigate how the congestion window dynamics and the links weights impact the system QoS performance for wireless networks. Our simulation results in [21] show that log-differential link weights, with a fixed congestion window size, reduce the file transfer delays. It will be certainly interesting to establish the validity of such an observation rigorously as a future research.

## References


[1] L. Tassiulas and A. Ephremides, Stability properties of constrained queueing systems and scheduling algorithms for maximal throughput in multihop radio networks, *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936-1948, 1992.

[2] X. Lin, N. Shroff, and R. Srikant, On the connection-level stability of congestion-controlled communication networks, *IEEE Transactions on Information Theory*, vol. 54, no. 5, pp. 2317-2338, 2008.

[3] J. Liu, A. Proutiere, Y. Yi, M. Chiang, and V. Poor, Flow-level stability of data networks with non-convex and time-varying rate regions, *Proc. ACM SIGMETRICS*, 2007.

[4] C. Moallemi and D. Shah, On the flow-level dynamics of a packet-switched network, *Proc. ACM SIGMETRICS*, pp. 83-94, June 2010.

[5] T. Bonald and M. Feuillet, On the stability of flow-aware CSMA, *Performance Evaluation*, vol. 67, no. 11, pp. 1219-1229, 2010.

[6] X. Lin, N. Shroff and R. Srikant, A tutorial on cross-layer optimization in wireless networks, *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1452-1463, 2006,

[7] M. J. Neely, E. Modiano, and C. E. Rohrs, Dynamic power allocation and routing for time varying wireless networks, *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 89-103, 2005.

[8] I. Keslassy and N. McKeown, Analysis of scheduling algorithms that provide 100% throughput in input-queued switches, *Proc. Allerton Conference on Communication, Control, and Computing*, 2001.

[9] A. Eryilmaz, R. Srikant, and J. R. Perkins, Stable scheduling algorithms for fading wireless channels. *IEEE/ACM Transactions on Networking*, vol. 13, no. 2, pp. 411-424, 2005.

[10] J. Ghaderi and R. Srikant, On the design of efficient CSMA algorithms for wireless networks, *IEEE Conference on Decision and Control*, 2010.

[11] J. Ni, B. Tan, R. Srikant, Q-CSMA: Queue-length based CSMA/CA algorithms for achieving maximum throughput and low delay in wireless networks, *Proc. IEEE INFOCOM Mini-Conference*, 2010.

[12] S. Rajagopalan, D. Shah and J. Shin, Network adiabatic theorem: an efficient randomized protocol for contention resolution, *ACM SIGMETRICS/Performance*, pp. 133-144, 2009.

[13] D. Shah and J. Shin, Randomized scheduling algorithm for queueing networks, *Annals of Applied Probability*, vol. 22, no. 1, pp. 128-171, 2011.

[14] P. Bremaud, Markov chains, Gibbs fields, Monte Carlo simulation, and queues, *Springer-Verlag*, New York 1999, 2nd edition, 2001.

[15] M. Crovella and A. Bestavros, Self-similarity in World Wide Web traffic: evidence and possible causes, *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 835-846, 1997.

[16] S. Asmussen, Applied probability and queues, Springer, 2003.

[17] L. Jiang and J. Walrand, A distributed CSMA algorithm for throughput and utility maximization in wireless networks, *46th Annual Allerton Conference on Communication, Control and Computing*, 2008.

[18] L. Jiang and J. Walrand, Convergence and stability of a distributed CSMA algorithm for maximal network throughput, *IEEE Conference on Decision and Control*, 2009.

[19] A. Proutiere, Y. Yi, and M. Chiang, Throughput of random access without message passing, *Proc. CISS*, Princeton, 2008.

[20] N. S. Walton, Utility optimization in congested queueing networks, *Journal of Applied Probability*, vol. 48, no. 1, pp. 68-89, 2011.

[21] J. Ghaderi, T. Ji, and R. Srikant, Connection-level scheduling in wireless networks using only MAC-layer information, *Proc. IEEE INFOCOM 2012 Mini-Conference*.

[22] D. P. Bertsekas and R. G. Gallager, Data Networks, *Prentice Hall*, 2nd edition, 1992

[23] J. Ghaderi and R. Srikant, Flow-level stability of multihop wireless networks using only MAC-layer information, *Proc. WiOpt 2012*.

[24] T. Ji and R. Srikant, Scheduling in wireless networks with connection arrivals and departures, *Information Theory and Applications Workshop*, 2011.

[25] J. Ghaderi and R. Srikant, Flow-level stability of multihop wireless networks: Separation of congestion control and packet scheduling, *Technical Report*, arXiv:1209.5464.


## Appendix A
## Proof of Lemma 1

Let $\hat{A}_{nf}^{(d)}(t)$ denote the number of packets of file $f$ injected into the MAC layer of node $n$, and $\hat{D}_{nf}^{(d)}(t) = \sigma_{nf}(t)I_{nf}(t)$ denote the expected "packet departure" of file $f$ from the

Transport layer. Let $B_{nf}(t) = \hat{A}^{(d)}_{nf}(t) - \hat{D}^{(d)}_{nf}(t)$ for file $f$. From the definition of $B_n(t)$, we have

$$B_n(t) = \sum_{f=1}^{N_n(t)} B_{nf}(t) + \sum_{f=N_n(t)+1}^{N_n(t)+a_n(t)} B_{nf}(t).$$

*Part (i):* It suffices to show that for each individual file $1 \le f \le N_n(t)$, $\mathbb{E}_{\mathcal{S}(t)}\Big[B_{nf}(t)\Big] = 0$. We only need to focus on files $f$ with $\xi_{nf}(t) = 1$, i.e., existing files in the Transport layer, or new files, i.e, $f \in \big(N_n(t)+1, N_n(t)+a_n(t)\big)$, because $\mathbb{E}_{\mathcal{S}(t)}\Big[B_{nf}(t)\Big] = 0$ if file $f$ has no packets in the Transport layer.

Let $\mathcal{W}^r_{nf}(t)$ be the remaining window size of file $f$ at node $n$ after MAC-layer departure but before the MAC-layer injection. We want to show that, for any $w \ge 0$,

$$\mathbb{E}_{\mathcal{S}(t)}\Big[B_{nf}(t)\Big|\mathcal{W}^r_{nf}(t) = w\Big] = 0, \tag{45}$$

then (45) implies $\mathbb{E}_{\mathcal{S}(t)}\Big[B_{nf}(t)\Big] = 0$. Because the number of remaining packets at the Transport layer at each time is geometrically distributed with mean size $\sigma_{nf}(t)$, the Transport layer will continue to inject packets into the MAC layer with probability $\varsigma_{nf}(t) = 1 - 1/\sigma_{nf}(t) = 1 - \eta_{nf}(t)$ as long as all previous packets are successfully injected and the window size is not full.

Clearly, if $w = 0$, no packet can be injected into the MAC layer. Therefore, $\hat{A}^{(d)}_{nf}(t) = 0$ and $\hat{D}^{(d)}_{nf}(t) = 0$, and (45) is satisfied. Next, consider the case $w > 0$. Let

$$p_w(k,j) := \mathbb{P}\left(\hat{A}^{(d)}_{nf}(t) = k, I_{nf}(t) = j|\mathcal{W}^r_{nf}(t) = w\right),$$

for $j \in \{0,1\}$ and $k \ge 1$. For $k \le w$, $p_w(k,1)$ directly follows the geometric distribution of the remaining packets of file $f$, i.e., for $1 \le k \le w$,

$$\begin{aligned} p_w(k,1) &= \mathbb{P}\left(\hat{A}^{(d)}_{nf}(t) = k|\mathcal{W}^r_{nf}(t) = w\right) \\ &= \varsigma^{k-1}_{nf}(t)(1-\varsigma_{nf}(t)). \end{aligned}$$

Note that from the definition of $I_{nf}(t)$, we have

$$\mathbb{P}\left(I_{nf}(t) = 0|\mathcal{W}^r_{nf}(t) = w\right) = 1 - \sum_{k=1}^{w} p_w(k,1) = \varsigma^w_{nf}(t).$$

Then, a simple calculation shows that

$$\begin{aligned} &\mathbb{E}_{\mathcal{S}(t)}\Big[B_{nf}(t)|\mathcal{W}^r_{nf}(t) = w\Big] = \sum_{k=1}^{w} p_w(k,1)\Big(k - \sigma_{nf}\Big) \\ &+ \mathbb{P}\left(I_{nf}(t) = 0|\mathcal{W}^r_{nf}(t) = w\right) w \\ &= \sum_{k=1}^{w} k\varsigma^{k-1}_{nf}(1-\varsigma_{nf}) - (1-\varsigma^w_{nf})\sigma_{nf} + w\varsigma^w_{nf} = 0, \end{aligned}$$

because $\varsigma_{nf} = 1 - 1/\sigma_{nf}$ by definition.

*Part (ii):* Using the fact that new arriving files are mutually independent, and are also independent of current network state, we can write $\mathbb{E}_{\mathcal{S}(t)}\Big[B_n(t)^2\Big] =$ "G" + "H" with

$$\text{“G”} := \mathbb{E}_{\mathcal{S}(t)}\Big[\Big(\sum_{f=1}^{N_n(t)} B_{nf}(t)\Big)^2\Big],$$
$$\text{“H”} := \mathbb{E}_{\mathcal{S}(t)}\Big[\sum_{f=N_n(t)+1}^{N_n(t)+a_n(t)} B_{nf}(t)^2\Big],$$

where we have also used the fact that $\mathbb{E}_{\mathcal{S}(t)}\Big[B_{nf}(t)\Big] = 0$. Note that $B_{nf}(t)^2 \le \max\{\hat{A}^{(d)}_{nf}(t)^2, \hat{D}^{(d)}_{nf}(t)^2\}$. Since the congestion window size is bounded by $\mathcal{W}_{cong}$ and the mean file size is bounded by $1/\eta_{min}$, we get $\mathbb{E}_{\mathcal{S}(t)}\Big[B_{nf}(t)^2\Big] \le \max\{\mathcal{W}^2_{cong}, 1/\eta^2_{min}\}$. Thus

$$\text{“H”} < \kappa_n \max\{\mathcal{W}^2_{cong}, 1/\eta^2_{min}\}.$$

Next, we bound the "G" term. Let $\mathcal{F}_n(t)$ denote the set of files at node $n$ that are served at time $t$. Because $B_{nf}(t) = 0$ if the existing file is not served, we have

$$\begin{aligned} \Big|\sum_{f=1}^{N_n(t)} B_{nf}(t)\Big| &\le \max\Big\{\sum_{f\in\mathcal{F}_n(t)} \hat{A}^{(d)}_{nf}(t), \sum_{f\in\mathcal{F}_n(t)} \sigma_{nf}(t)\Big\} \\ &\le \big|\mathcal{F}_n(t)\big| \cdot \max\Big\{\mathcal{W}_{cong}, 1/\eta_{min}\Big\}. \end{aligned}$$

Note that $|\mathcal{F}_n(t)| \le \sum_{j:(n,j)\in\mathcal{L}} x_{nj}(t) \le Nr_{max}$ because the number of existing files that are served cannot exceed the sum of outgoing link capacities. Thus,

$$\text{“G”} \le N^2 r^2_{max} \max\Big\{\mathcal{W}^2_{cong}, 1/\eta^2_{min}\Big\}.$$

This completes the proof.

## Appendix B
## Proof of Lemma 2

Note that $u^{(d)}_n(t) = 0$ if $q^{(d)}_n(t) \ge Nr_{max}$, and $u^{(d)}_n(t) \le Nr_{max}$ if $q^{(d)}_n(t) \le Nr_{max}$. In the latter case, since the congestion window size for every file is at least one, there are at most $Nr_{max}$ files in the Transport layer of node $n$ intended for destination $d$. Hence, using the definition of $\bar{Q}^{(d)}_n(t)$, $\bar{Q}^{(d)}_n(t) \le Q^0 := Nr_{max} + Nr_{max}/\eta_{min}$. So,

$$\begin{aligned} &\mathbb{E}_{\mathcal{S}(t)}\Big[g(\bar{Q}^{(d)}_n(t))u^{(d)}_n(t)\Big] = \\ &\mathbb{E}_{\mathcal{S}(t)}\Big[g(\bar{Q}^{(d)}_n(t))u^{(d)}_n(t)\mathbb{1}\Big\{q^{(d)}_n(t) \le Nr_{max}\Big\}\Big] \le \\ &\mathbb{E}_{\mathcal{S}(t)}\Big[g(\bar{Q}^{(d)}_n(t))Nr_{max}\mathbb{1}\Big\{q^{(d)}_n(t) \le Nr_{max}\Big\}\Big] \le \\ &Nr_{max}g(Q^0). \end{aligned}$$

Therefore $C_2 = N^3 r_{max} g(Nr_{max}(1 + 1/\eta_{min}))$.

## Appendix C
## Proof of Lemma 6

Note that

$$\frac{\pi_{t+1}(s)}{\pi_t(s)} = \frac{Z_t}{Z_{t+1}} \exp\Big(\sum_{(i,j)\in s} (\widetilde{w}_{ij}(t+1) - \widetilde{w}_{ij}(t))\Big),$$

where

$$\begin{aligned} \frac{Z_t}{Z_{t+1}} &= \frac{\sum_{s\in\mathcal{R}} \exp(\sum_{(i,j)\in s} \widetilde{w}_{ij}(t))}{\sum_{s\in\mathcal{R}} \exp(\sum_{(i,j)\in s} \widetilde{w}_{ij}(t+1))} \\ &\le \max_s \exp\Big(\sum_{(i,j)\in s} (\widetilde{w}_{ij}(t) - \widetilde{w}_{ij}(t+1))\Big) \\ &\le \exp\Big(\sum_{(i,j)\in\mathcal{L}} (\widetilde{w}_{ij}(t) - \widetilde{w}_{ij}(t+1))\Big). \end{aligned}$$

Let $q^*(t)$ denote $g^{-1}(g^*(t))$, and define $\widetilde{q}_i^{(d)}(t) := \max\{q^*(t), q_i^{(d)}(t)\}$. Then,

$$\widetilde{w}_{ij}^{(d)}(t+1) - \widetilde{w}_{ij}^{(d)}(t) =$$
$$g(\widetilde{q}_i^{(d)}(t+1)) - g(\widetilde{q}_j^{(d)}(t+1)) - g(\widetilde{q}_i^{(d)}(t)) + g(\widetilde{q}_j^{(d)}(t)) =$$
$$\left[g(\widetilde{q}_i^{(d)}(t+1)) - g(\widetilde{q}_i^{(d)}(t))\right] + \left[g(\widetilde{q}_j^{(d)}(t)) - g(\widetilde{q}_j^{(d)}(t+1))\right].$$

Recall that the link service rate is at most one and the congestion window sizes are at most $\mathcal{W}_{cong}$, thus $\forall i \in \mathcal{N}$, $\forall d \in \mathcal{D}$, $|\widetilde{q}_i^{(d)}(t+1) - \widetilde{q}_i^{(d)}(t)| \leq 1 + \mathcal{W}_{cong}$. Hence,

$$\begin{aligned}\frac{|\widetilde{w}_{ij}^{(d)}(t+1) - \widetilde{w}_{ij}^{(d)}(t)|}{1+\mathcal{W}_{cong}} &\leq g'(\widetilde{q}_i^{(d)}(t)) + g'(\widetilde{q}_j^{(d)}(t+1)) \\ &\leq 2g'(q^*(t+1) - 1 - \mathcal{W}_{cong}),\end{aligned}$$

where we have also used the fact that $g$ is a concave increasing function. Therefore,

$$\frac{\pi_{t+1}(s)}{\pi_t(s)} \leq e^{2(1+\mathcal{W}_{cong})|\mathcal{L}|g'(q^*(t+1)-1-\mathcal{W}_{cong})}.$$

A similar calculation shows that also

$$\frac{\pi_t(s)}{\pi_{t+1}(s)} \leq e^{2(1+\mathcal{W}_{cong})|\mathcal{L}|g'(q^*(t+1)-1-\mathcal{W}_{cong})}.$$

This concludes the proof.

## Appendix D
## Proof of Lemma 7

The second inequality immediately follows from definition of $w_{ij}$. To prove the first inequality, consider a destination $d$, with routing matrix $\mathbf{R}^{(d)} \in \{0,1\}^{N\times N}$, and let $\mathbf{w}^{(d)} = [w_{ij}^{(d)}(t) : R_{ij}^{(d)} = 1]$, then, based on (2), we have

$$\mathbf{w}^{(d)} = (\mathbf{I} - \mathbf{R}^{(d)})g(\mathbf{q}^{(d)}),$$

where $g(\mathbf{q}^{(d)}) = [g(q_i^{(d)}) : i \in \mathcal{N}]$. Note that every row of $\mathbf{R}^{(d)}$ has exactly one "1" entry except the row corresponding to $d$ which is all zero, so $(\mathbf{R}^{(d)})^N = 0$. Therefore, $(\mathbf{I}-\mathbf{R}^{(d)})^{-1} = \mathbf{I}+\mathbf{R}^{(d)}+(\mathbf{R}^{(d)})^2+\cdots$ exists and $\mathbf{I}-\mathbf{R}^{(d)}$ is nonsingular. So $g(\mathbf{q}^{(d)}) = (\mathbf{I}-\mathbf{R}^{(d)})^{-1}\mathbf{w}^{(d)}$. Let $\|\cdot\|_\infty$ denote the $\infty$-norm. Then we have

$$\begin{aligned}\|(\mathbf{I}-\mathbf{R}^{(d)})^{-1}\|_\infty &= \|\sum_{k=0}^{N}(\mathbf{R}^{(d)})^k\|_\infty \leq \sum_{k=0}^{N}\|(\mathbf{R}^{(d)})^k\|_\infty \\ &\leq \sum_{k=0}^{N}\|\mathbf{R}^{(d)}\|_\infty^k \leq N\end{aligned}$$

where we have used the basic properties of the matrix norm, and the fact that $\|\mathbf{R}^{(d)}\|_\infty = 1$. Therefore,

$$\|g(\mathbf{q}^{(d)})\|_\infty \leq \|(\mathbf{I}-\mathbf{R}^{(d)})^{-1}\|_\infty\|\mathbf{w}^{(d)}\|_\infty \leq N\|\mathbf{w}^{(d)}\|_\infty,$$

for every $d \in \mathcal{D}$. Taking the maximum over all $d \in \mathcal{D}$, and noting that $g$ is a strictly increasing function, yields the result.

## Appendix E
## Proof of Lemma 8

$h$ is strictly increasing so $h(x) \geq 1$ for all $x \geq h^{-1}(1)$. So $g'(x) \leq \frac{1}{1+x}$ for $x \geq h^{-1}(1)$. The inverse of $g$ cannot be expressed explicitly, however, it satisfies

$$g^{-1}(x) = \exp(xh(g^{-1}(x))) - 1. \tag{46}$$

Therefore,

$$\begin{aligned}\alpha_t &\leq \frac{2(1+\mathcal{W}_{cong})|\mathcal{L}|}{g^{-1}(g^*) - \mathcal{W}_{cong}} &(47)\\ &= \frac{2(1+\mathcal{W}_{cong})|\mathcal{L}|}{\exp(g^*h(g^{-1}(g^*))) - 1 - \mathcal{W}_{cong}}. &(48)\end{aligned}$$

for $g^* \geq g(1+\mathcal{W}_{cong}+h^{-1}(1))$. Next, note that

$$T_{t+1} \leq 16^{|\mathcal{L}|}e^{4|\mathcal{L}|(w_{max}+g^*)} \leq$$
$$16^{|\mathcal{L}|}e^{4|\mathcal{L}|(g(q_{max})+\frac{\epsilon}{4|\mathcal{L}|N}g(q_{max}))} \leq 16^{|\mathcal{L}|}e^{8|\mathcal{L}|g(q_{max})}. \tag{49}$$

Consider the product of (48) and (49) and let $K := 2(\mathcal{W}_{cong}+1)|\mathcal{L}|16^{|\mathcal{L}|}$. Using (46) and (27), the condition (35) is satisfied if

$$Ke^{g^*[\frac{32|\mathcal{L}|N^3}{\epsilon} - h(g^{-1}(g^*))]}\left(1 + \frac{1+\mathcal{W}_m}{g^{-1}(g^*) - \mathcal{W}_m}\right) \leq \delta/16. \tag{50}$$

Consider fixed, but arbitrary, $|\mathcal{L}|$, $N$ and $\epsilon$. As $q_{max} \to \infty$, $g(q_{max}) \to \infty$, and consequently $g^* \to \infty$ and $g^{-1}(g^*) \to \infty$. Therefore, the exponent $\frac{32|\mathcal{L}|N^3}{\epsilon} - h(g^{-1}(g^*))$ is negative for $q_{max}$ large enough, and thus, there is a threshold $q_{th}$ such that for all $q_{max} > q_{th}$, the condition (50) is satisfied.

The last step of the proof is to determine $t^*$. Let $t_1$ be the first time that $q_{max}(t)$ hits $q_{th}$, then

$$\begin{aligned}\sum_{k=t_1}^{t_1+t}\frac{1}{T_k^2} &\geq 16^{-2|\mathcal{L}|}\sum_{k=t_1}^{t_1+t}e^{-16|\mathcal{L}|g(q_{max}(t))} \\ &= 16^{-2|\mathcal{L}|}\sum_{k=t_1}^{t_1+t}(1+q_{max}(t))^{-\frac{16|\mathcal{L}|}{h(q_{max}(t))}} \\ &\geq 16^{-2|\mathcal{L}|}t(1+q_{th}+t)^{-\frac{16|\mathcal{L}|}{h(q_{th})}},\end{aligned}$$

and

$$\begin{aligned}\min_s \pi_{t_1}(s) &\geq \frac{1}{\sum_s \exp(\sum_{i\in s}\widetilde{w}_{ij}(t_1))} \\ &\geq \frac{1}{|\mathcal{R}|\exp(|\mathcal{L}|(w_{max}(t_1)+g^*(t_1)))} \\ &\geq \frac{1}{2^{N^2}\exp(2N^2g(q_{th}))}.\end{aligned}$$

Therefore, by Proposition 1, it suffices to find the smallest $t$ that satisfies

$$\begin{aligned}16^{-2N^2}t(1+q_{th}+t)^{-\frac{16N^2}{g(q_{th})}} &\geq \log(4/\delta) \\ &+ N^2\log(2(1+q_{th}))\end{aligned}$$

for a threshold $q_{th}$ large enough. Recall that $h(.)$ is an increasing function, therefore, by choosing $q_{th}$ large enough, $\frac{16N^2}{h(q_{th})}$ can be made arbitrary small. Then a finite $t^*$ always exists since $\lim_{t^*\to\infty} t^*(1+q_{th}+t^*)^{-\frac{16N^2}{h(q_{th})}} = \infty$.