MaxWeight vs. BackPressure: Routing and Scheduling in Multi-Channel Relay Networks

Sharayu Moharir and Sanjay Shakkottai, *Fellow, IEEE*

***Abstract*—We study routing and scheduling algorithms for relay-assisted, multi-channel downlink wireless networks (e.g., OFDM-based cellular systems with relays). Over such networks, while it is well understood that the BackPressure algorithm is stabilizing (i.e., queue lengths do not become arbitrarily large), its performance (e.g., delay, buffer usage) can be poor. In this paper, we study an alternative – the MaxWeight algorithm – variants of which are known to have good performance in a single-hop setting. In a general relay setting however, MaxWeight is not even stabilizing (and thus can have very poor performance).**

In this paper, we study an iterative MaxWeight algorithm for routing and scheduling in downlink multi-channel relay networks. We show that, surprisingly, the iterative MaxWeight algorithm can stabilize the system in several large-scale instantiations of this setting (e.g., general arrivals with full-duplex relays, bounded arrivals with half-duplex relays). Further, using both many-channel large-deviations analysis and simulations, we show that iterative MaxWeight outperforms the BackPressure algorithm from a queue-length/delay perspective.

***Index Terms*—Wireless Scheduling and Routing, Downlink Relay Networks.**

I. Introduction

We consider OFDM (Orthogonal Frequency Division Multiplexing) based multichannel multihop downlink networks consisting of a base-station, relays and users. OFDM based networks are widely being deployed in commercial cellular networks (e.g., LTE [1]); looking forward, it is well recognized that wireless relays are envisioned to be an integral part of the solution for next generation cellular systems (e.g., LTE-Advanced [12]). The setting here – multichannel OFDM wireless networks – is the de-facto standard for 4G cellular communications. These systems have several tens of parallel channels (e.g., WiMax over 20 MHz bandwidth has about 50 channels, with each channel having 25 OFDM sub-carriers grouped together) [3], [4]. A key challenge here is to design good routing and scheduling algorithms that provide good user performance (e.g., small buffer usage, low delay, etc.)

The obvious candidate for scheduling and routing in this scenario is the BackPressure algorithm [20], which routes and schedules packets based on *differential backlogs* (i.e., queue-length differences from a one-hop downstream node). This algorithm is known to be stabilizing; however, it is known that it can have poor delay performance [24], [6], [19]. An alternative, which simply looks at backlogs and *not* differential backlogs is the MaxWeight algorithm [21]. The MaxWeight algorithm assigns a weight of (queue-length $\times$ channel-rate), and schedules a collection of links that maximizes the total weight (max-weight independent set). This algorithm is however, *not* stabilizing in general, and thus results in very poor performance. As a simple example, we study the 4-node network in Figure 1, where the source node (A) needs to deliver packets at rate 1.5 packets/slot to the destination (D). The only scheduling constraint is that links l_1 and l_2 cannot be activated together. It is clear that with the MaxWeight algorithm, the source node A *always* routes packets along link l_1 (with capacity of 10 packets/slot) and does not utilize the lower path (see figure) due to the scheduling constraint (because the weight of the link l_1 is always 10 times larger than the weight of l_2). This results in the buffer at node B becoming arbitrarily large (as the corresponding outgoing link can only support 1 packet/slot). This example seems to indicate that MaxWeight is not a good candidate for relay network scheduling and routing. Surprisingly, in this paper, we show that the above intuition is *not true* in large-scale downlink networks. We show that for large enough multi-channel downlink relay networks, MaxWeight type algorithms do stabilize the system and have better buffer-usage performance than the BackPressure algorithm. Such smaller buffer usage leads to a corresponding smaller packet delay. The intuition that leads to these results is that in networks with a large number of channels (multiple OFDM channels), *(i)* there is sufficient flexibility due to the *degrees of freedom* that the channels provide that can compensate for routing inefficiencies in MaxWeight, and *(ii)* by not considering downlink backlogs, upstream nodes with the MaxWeight algorithm are more aggressive in using good channels to “push” packets closer toward the destination, and thus resulting in better overall performance than BackPressure.

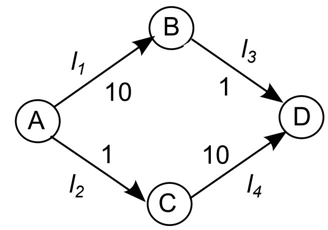

Fig. 1. A relay network (Example 1) illustrating that MaxWeight algorithm is not stabilizing. There are four links (l_1, l_2, l_3, l_4) with capacities being $(10, 1, 1, 10)$ packets/slot respectively. The source node is A and the destination is D.

This work was partially supported by NSF grants CNS-0964391, CNS-1017549, CNS-1161868 and CNS-1343383.

An earlier version of this paper appeared in the Proceedings of INFOCOM, 2013.

The authors are with the Department of Electrical and Computer Engineering at the University of Texas at Austin, TX 78712, USA.

A. Related Work

Performance with MaxWeight and BackPressure algorithms has been studied in many settings over the last decade. With fixed routing (including single-hop flows), delay and buffer-size performance has been studied for mean delay [15], [7] and large buffer asymptotes [25], [22], [16], [18], [23]. Also, from a network stability viewpoint for MaxWeight, work includes [9] where the authors show that the network is stable if the routes are fixed, and nodes are "decoupled" by means of "measuring" arrival rates [11]. In this work, we focus on properties (stability and queue-length/delay performance) of variants of the BackPressure and MaxWeight algorithms for networks which require dynamic routing.

With dynamic routing and BackPressure like algorithms, modifications have been proposed to queue structures (e.g, shadow queue [6], virtual queues [8], per-hop queues [24]) that empirically result in lower end-to-end delay. Closer to our setting with multiple channels (but only single-hop downlink), large deviation analysis provides buffer-size [3], [4], [5] or delay [17], [10] performance bounds for iterative algorithms.

Our focus here is on downlink multi-hop networks – in this setting, MaxWeight algorithms for *routing* have not been studied (either in single-channel or multi-channel settings) as these algorithms are believed to be not even stabilizing.

B. Contributions

We propose four routing and scheduling algorithms called the Server Side Greedy (SSG) BackPressure algorithm, the SSG MaxWeight algorithm, the Iterative Longest Queue First (ILQF) BackPressure algorithm and the ILQF MaxWeight algorithm in Section IV. We show the following:

1) BackPressure Algorithm:

- We prove that the BackPressure algorithm does not have good small-queue performance. We show that rate function of the maximum queue length is zero for i.i.d. ON-OFF channels, i.i.d. Bernoulli arrivals, and linear scaling of the number of relays.

2) SSG BackPressure Algorithm:

- The algorithm is throughput optimal for the 2-hop networks we consider under general arrival processes, and bounded channel processes.

3) SSG MaxWeight Algorithm:

- For 2-hop downlink networks, for arrival rate vectors strictly in the interior the stability region of the system that satisfy some additional constraints, if the system scale is large enough, the algorithm keeps the system stable (see Section V for specific details).
- For i.i.d. ON-OFF channels, i.i.d. Bernoulli arrivals and linear scaling of the number of relays, we show that the maximum queue length rate function is strictly positive (i.e., exponential decay in queue length tails).

4) ILQF MaxWeight and ILQF BackPressure Algorithms:

- For i.i.d. ON-OFF channels, i.i.d. Bernoulli arrivals and linear scaling of the number of relays, we show that the maximum queue length rate function is strictly positive (i.e., exponential decay in queue length tails).

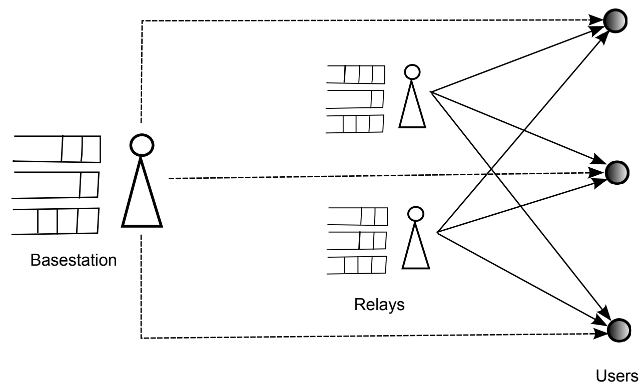

Fig. 2. An illustrative example of a 2-hop relay network with 2 relays and 3 users.

We compare the lower bounds on the rate functions of the SSG MaxWeight algorithm, the ILQF MaxWeight algorithm and the ILQF BackPressure algorithm and compare their delay performance via simulations. In particular, the bounds for the MaxWeight based algorithms are greater than the bounds for the BackPressure based algorithm and our simulations verify these results.

We finally note that while we have stated and proved the results in the context of 2-hop networks, the results can be easily extended to any k-hop downlink network (i.e., multiple "layers" of relays). We skip the details to keep notation manageable.

II. System Model: 2-Hop Downlink Communication Networks

We consider a multiuser, multichannel 2-hop downlink communication system. The system consists of a base-station (BS), $R(n)$ relays and n users and n channels, the base-station and the relays maintain n queues each, one for each user in the system as shown in Figure 2.

Our results can be generalized to the case where the two quantities (number of users and number of channels) are not equal, but scale linearly with respect to each other. We consider the case when the two are equal to keep the notation simple.

We study a discrete time queuing system. We build on the notation used in [3], [4], [5]. All queue lengths below (i.e., at the BS and relays) are measured at the end of a time-slot t, and arrivals occur at the beginning of the time-slot.

- Q_i = Queue number i at the base-station.
- R_{ri} = Queue number i at relay r.
- S_i = Channel number i.
- $Q_i(t)$ = The queue length of user i at the BS (measured at the end of the time-slot).
- $\mathbf{Q}(t) = \{Q_i(t) : 1 \leq i \leq n\}$: The vector of queue lengths at the base-station.
- $R_{ri}(t)$ = The queue length of user i at relay r (measured at the end of the time-slot).
- $\mathbf{R}(t) = \{R_{ri}(t) : 1 \leq r \leq R(n), 1 \leq i \leq n\}$: The vector of queue lengths at the relays.
- $A_i(t)$ = The number of arriving packets to Q_i at the base-station.

- $\mathbf{A}(t) = \{A_i(t) : 1 \le i \le n\}$: The vector of the number of arriving packets at the base-station at the beginning of time-slot t .
- $A_i^r(t)$ = The number of arriving packets to R_{ri} (measured at the beginning of the time-slot).
- $X_{i,j}(t)$ = The number of packets in Q_i that can be transmitted by the BS to user i on channel j in time-slot t.
- $X_{i,j}^{B,r}(t)$ = The number of packets in Q_i that can be transmitted by the BS to relay r on channel j in time-slot t.
- $X_{i,j}^{r}(t)$ = The number of packets in R_{ri} that can be transmitted by the relay r to user i on channel j in time-slot t.

Note that arrivals to the base-station queues are external and the arrivals to the relay queues are intermediate, i.e., packets sent from the base-station to the relays. We design algorithms that assign channels to the base-station and relay queues in every time-slot, and execute their allocation through the variables $Y_{i,j}^{B,r}(t)$, $Y_{i,j}^{r}(t)$ and $Y_{i,j}(t)$ for $1 \le i \le n, 1 \le j \le n$ and $1 \le r \le R(n)$. These variables are defined as follows:

- $Y_{i,j}(t)$ is 1 if channel j is scheduled for transmission from Q_i to user i in time-slot t and 0 otherwise.
- $Y_{i,j}^{B,r}(t)$ is 1 if channel j is scheduled for transmission from Q_i to R_{ri} in time-slot t and 0 otherwise.
- $Y_{i,j}^{r}(t)$ is 1 if channel j is scheduled to serve the queue for user i at relay r in time-slot t and 0 otherwise.

The dynamics of the individual queues in the system is described below:

$$\begin{aligned} Q_i(t) &= \Bigg(Q_i(t-1) + A_i(t) \\ &\quad - \sum_{j=1}^{n}\sum_{r=1}^{R(n)} X_{i,j}^{B,r}(t)Y_{i,j}^{B,r}(t) - \sum_{j=1}^{n} X_{i,j}(t)Y_{i,j}(t)\Bigg)^+, \\ R_{ri}(t) &= \Bigg(R_{ri}(t-1) + A_i^r(t) - \sum_{j=1}^{n} X_{i,j}^{r}(t)Y_{i,j}^{r}(t)\Bigg)^+, \end{aligned}$$

where

$$A_i^r(t) = \text{the number of packets for user } i \text{ received by relay } r \text{ at the beginning of time-slot } t.$$

We consider the following Interference Models:

1) Full Duplex: In the full duplex model, each relay has two transceivers and therefore, can receive and transmit on the same channels simultaneously.
2) Half Duplex: In the half duplex model, the relays can either receive or transmit in a time-slot.

Using these two interference models, it is possible to construct multiple types of Multihop relay networks. For instance:

1) **Full Duplex without Direct Link (FD-w/oDL)**
In this model, we assume that the relays are full duplex and there is no direct communication link between the base-station and the users. We assume that the interference graph for the relays is a complete graph, i.e., only one of the relays can transmit on a particular channel in a give slot.
2) **Full Duplex with Direct Link (FD-wDL)**
In this model, we assume that the relays are full duplex and there is a direct communication link between the base-station and the users. We assume that the interference graph for the relays is a complete graph.
3) **Half Duplex with Direct Link (HD-wDL)**
In this model, we assume that the relays are half duplex and there is a direct communication link between the base-station and the users. We assume that the interference graph for the relays is a complete graph.

For our results, the interference graph of the relays being a complete graph is the most restrictive condition that can be imposed on interference among the relays. We can show that the same results apply for less restrictive interference constraints. However, we skip the details for brevity. In this paper, we look at the FD-w/oDL and HD-wDL Models in detail. The results and proofs for FD-w/oDL similarly extend to the FD-wDL Model.

III. Background: The SSG Scheduling Algorithm

In this section we discuss the Server Side Greedy (SSG) algorithm proposed in [4] which is known to have good delay performance for single hop downlink networks.
The Server Side Greedy (SSG) algorithm was defined in [4] for a single hop downlink system. This algorithm sequentially allocates channels to queues within each time-slot. It first allocates channel S_1 to the maximum weight queue, i.e., the queue with largest $(Q_i(t)X_{i,1}(t))$. It updates the queue length based on the number of packets that are drained due to this allocation, and proceeds sequentially to the next channel (and so on). The key point is that even within a time-slot, queue lengths are updated during the allocation process, and future channel allocations within the time-slot take the accumulated queue length drains into account. For a formal definition of the SSG algorithm (and proofs that this has quadratic complexity in n), please refer to [4], Definition 3.

IV. Proposed Scheduling and Routing Algorithms for 2-Hop Downlink Networks

The SSG algorithm discussed in Section III was designed for single hop networks and therefore designed only for scheduling packets. In this section, we build on the SSG algorithm to design scheduling and routing algorithms for multihop downlink networks. We describe the algorithms in the context of 2-hop networks for simplicity, but, they can be extended to k-hop downlink networks.

A. FD-w/oDL Model

Input:

- The queue lengths $Q_i(t-1)$ and $R_{ri}(t-1)$, for $1 \le i \le n$, $1 \le r \le R(n)$.
- The arrival vectors $A_i(t)$ and $A_i^r(t)$, for $1 \le i \le n$, $1 \le r \le R(n)$.
- The channel realizations $X_{i,j}^{r}(t)$ and $X_{i,j}^{B,r}(t)$ for $1 \le i \le n$, $1 \le j \le n$, $1 \le r \le R(n)$.

1) SSG BackPressure for FD-w/oDL: The allocation for relay queues is carried out first using the SSG rule (tie breaking rule: highest priority is the smallest relay index followed by the smallest user index). The updated relay queue lengths are used for allocation of channels at the BS using the SSG rule with the weight of each link being the backpressure-channel product of that link (tie breaking rule: highest priority is the smallest relay index followed by the smallest user index at each relay).

2) SSG MaxWeight for FD-w/oDL: The allocation for relay queues is carried out first using the SSG rule (tie breaking rule: highest priority is the smallest relay index followed by the smallest user index). The allocation for the BS queues is also done using the SSG rule with the weight of each link being the queue-length-channel product of that link, breaking ties in a cyclic order as follows. We initialize the priority order of the relays as $\{1, 2, .., R(n)\}$. In each round of the allocation process, the relay that is allocated that particular channel is then removed from its current position in the priority order and inserted at the last position to get the new priority order.

B. HD-wDL Model

Input:

- The queue lengths $Q_i(t-1)$ and $R_{ri}(t-1)$, for $1 \leq i \leq n$, $1 \leq r \leq R(n)$.
- The arrival vectors $A_i(t)$ and $A_i^r(t)$, for $1 \leq i \leq n$, $1 \leq r \leq R(n)$.
- The channel realizations $X_{i,j}^r(t)$, $X_{i,j}^{B,r}(t)$ and $X_{i,j}(t)$ for $1 \leq i \leq n$, $1 \leq j \leq n$, $1 \leq r \leq R(n)$.

1) SSG BackPressure for HD-wDL Model: Let

$$\begin{aligned}\Delta\xi_B(t-1) &= \max_{1\leq i\leq n, 1\leq r\leq R(n)} \\ &\quad (Q_i(t-1) - R_{ri}(t-1) + A_i(t)), \\ \xi_R(t-1) &= \max_{1\leq i\leq n, 1\leq r\leq R(n)} (R_{ri}(t-1) + A_i^r(t)).\end{aligned}$$

If $\Delta\xi_B(t-1) > \xi_R(t-1)$, the base-station queues transmit in slot t, else the relay queues transmit in slot t. The allocation for relay queues is carried out using the SSG rule (tie breaking rule: highest priority is the smallest relay index followed by the smallest user index). The allocation for the BS queues is done using the SSG rule with the weight of each link being the backpressure-channel product of that link (tie breaking rule: highest priority is the smallest relay index followed by the smallest user index).

2) SSG MaxWeight for HD-wDL Model: Initialize

$$A_{max} = \max_{1\leq i\leq n} A_i(0).$$

In each time-slot t, update

$$A_{max} = \max\left\{A_{max}, \max_{1\leq i\leq n} A_i(t)\right\}.$$

Let

$$\begin{aligned}\xi_B(t-1) &= \max_{1\leq i\leq n} (Q_i(t-1) + A_i(t)), \\ \xi_R(t-1) &= \max_{1\leq i\leq n, 1\leq r\leq R(n)} (R_{ri}(t-1) + A_i^r(t)).\end{aligned}$$

If $\xi_B(t-1) > \xi_R(t-1)$, the base-station queues transmit in slot t, else the relay queues transmit in slot t. The allocation for relay queues is carried out using the SSG rule (tie breaking rule: highest priority is the smallest relay index followed by the smallest user index). The allocation for the BS queues is also done using the SSG rule till all queues have queue length less than $\xi_B(t-1) - A_{max} - 1$ or we run out of channels to allocate.

V. Main Results and Discussion

We now state our main results, and discuss their implications.

A. Stability

Assumption 1: We use similar Assumptions to [7], [4], described below for completeness.

1) The channel process:
   - The channel state process is assumed to have a stationary distribution $\pi = [\pi]_{i\in I}$, with $\pi_i > 0$ for all $i \in I$ where I is the collection of possible channel states.
   - Denote $s[m]$ to be the channel state in time-slot m. We assume that for any $\epsilon > 0$, there exists an integer $M_0 > 0$ such that for all $M \geq M_0$, all $i \in I$, and all k, we have
   $$E\left[\left|\pi_i - \frac{1}{M}\sum_{m=k}^{k+M-1} 1_{s[m]=i}\right|\right] < \epsilon.$$
   - There exists $X_{max} > 0$ such that
   $$\max_{i,j,t} X_{ij}(t) \leq X_{max}.$$
2) The arrival process:
   - The arrival process to each node n_i in the network is a stationary process with mean λ_i.
   - The arrival rates which lie in the interior of the system's throughput region.
   - Given any $\epsilon > 0$, we assume that there exists an integer $M_1 > 0$ such that for all $M \geq M_1$, and for all k, i,
   $$E\left[\left|\lambda_i - \frac{1}{M}\sum_{m=k}^{k+M-1} A_i(m)\right|\right] < \epsilon.$$
   - The second moment of the number of arrivals per time-slot is bounded.

For the following theorem, we consider the SSG BackPressure algorithm for any of the models described so far (i.e., FD-w/oDL, FD-wDL, HD-wDL). This theorem continues to hold for any multi-channel network with independent sets based scheduling constraints (in this case, the SSG BackPressure algorithm sequentially allocates max-weight independent sets).

Theorem 1 (Throughput Optimality of SSG BackPressure)**.** *Under Assumption 1, the SSG BackPressure rule results mean-stable queues, i.e.,*

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}\sqrt{\sum_{i=1}^{n} Q_i^2(t) + \sum_{i=1}^{n}\sum_{r=1}^{R(n)} R_{ri}^2(t)} < \infty.$$

As the name suggests, this algorithm takes into account previous channel and user allocations (and the changes in queue lengths due to such allocations) for each successive new channel allocation. The proof of this builds on techniques in [4], [7]. This result shows that the SSG BackPressure algorithm keeps the queues stable, and thus is a candidate for studying other performance measures such as buffer usage or delay. Please refer to [14] for the proof of this theorem.

Assumption 2: (FD-w/oDL: Stability)

- **Assumption 2(a)**:
  Arrivals and Bounded Channels
  - We assume that $\mathbf{A}(t)$ (the vector of arrivals in a time-slot across users) is an aperiodic, irreducible, finite state Markov chain (independent of the channel process).
  - We define $\lambda = \frac{1}{n} E\left[\sum_{i=1}^{n} A_i(0)\right]$. Then,
  $$P\left(\sum_{i=1}^{n} A_i(t) \geq n(\lambda + \delta)\right) \leq e^{-nk(\delta)},$$
  where $k(\delta) > 0$ is a function of δ and independent of n.
  - $A_i(t) \leq k_1 n$ for all t and i and some constant k_1.
  - The channel processes are i.i.d. across time-slots.
  - $X_{i,j}^{B,r}(t) \leq S_{max} < \infty$.
  - $X_{i,j}^{r}(t) \leq S_{max} < \infty$.
  - For every i, j, r and t,
  $$P(X_{i,j}^{r}(t) = S_{max}) = q(i,j,r) > 0.$$
- **Assumption 2(b)**:
  Consider the event E that there exists a set of channels J such that $|J| = nk_2$ for some constant $k_2 < 1$ and $X_{i,j}^{B,r} < S_{max}$ for all $j \in J$ and $1 \leq r \leq R(n)$. Then,
  $$P(E) = o\left(\frac{1}{n^6}\right).$$
  The event E as described above is equivalent to saying that in a given time-slot, there exists a constant fraction of the channels which cannot be used at S_{max} by the base-station. If the channels are i.i.d. Bernoulli with parameter q across relays and time, we have that
  $$P(E) = 2^{nH(k_2)}(1-q)^{nk_2R(n)} = o\left(\frac{1}{n^6}\right),$$
  where $H(k_2) = -(k_2\log(k_2) + (1-k_2)\log(1-k_2))$. We can show that another sufficient condition is the α mixing condition defined in [2]. The condition implies that even though the channel variables are not independent, the correlation between them decays over space and time and, α captures the rate at which correlation decays.
- **Assumption 2(c)**:
  Let I be a set of relays such that $|I| \geq \delta R(n)$, for some constant $\delta < 1$. Consider the event G that for a channel j and for every relay $r \in I$, $X_{i,j}^{r}(t) < S_{max}$, $\forall i$. Then,
  $$P(G) \leq o\left(\frac{1}{n^4}\right).$$
  If the channels are i.i.d. Bernoulli with parameter q across relays and time, for $\delta = 0.5$, we have that
  $$P(G) \leq (1-q)^{0.5R(n)}.$$
  Therefore, for i.i.d. channels, we need $R(n) > -\frac{6}{\log(1-q)}\log n$. The event G as described above is that given a set of relay which includes δ fraction of all the $R(n)$ relays, none of them can use a channel j at S_{max} in a given time-slot.
- **Assumption 2(d)**:
  Let I be a set of relay queues such that that $|I| = k_3R(n)$ for some constant $k_3 < 1$ and let J be a set of channels such that $|J| = \frac{2k_3R(n)}{q_{min}}$, where
  $$q_{min} = \min_{r,i,j,t} q(i,j,r,t) > 0.$$
  Consider the event W that for every relay in I there exist $k_3R(n)$ channels in J such that $X_{i,j}^{r}(t) = S_{max}$. Since $|J| = \frac{2k_3R(n)}{q_{min}}$, for every relay, the expected number of channels in J such $X_{i,j}^{r}(t) = S_{max}$ is at least $2k_3R(n)$. Therefore W is the event that for all relays, the number of channels which have rate S_{max} is at least half of its expected value. Then,
  $$P(W^c) = o\left(\frac{1}{n^3}\right).$$
  If the channels are i.i.d. Bernoulli with parameter q across relays and time, we have that
  $$P(W^c) = k_3R(n)e^{-\frac{2k_3R(n)}{q}H(\frac{q}{2}|q)}.$$
  This assumption, we can show, is also satisfied by the α mixing condition defined in [2] and discussed in Assumption 2(b).

Lemma 1. *Under Assumption 2, if* $\frac{1}{n}E\left[\sum_{i=1}^{n} A_i(0)\right] = \lambda > S_{max}$*, no scheduling algorithm can stabilize the system.*

Therefore, $\lambda \leq S_{max}$ is a necessary condition for an arrival vector to lie in the stability region of the system.

Theorem 2. *Under Assumption 2, for arrival processes with* $\lambda < S_{max}$*, the SSG MaxWeight algorithm stabilizes the FD-w/oDL system, i.e., the markov chain* $\{\boldsymbol{Q}(t), \boldsymbol{R}(t), \boldsymbol{A}(t)\}$ *is positive recurrent for* $n > n_0$ *where* n_0 *is a function of* λ.

This is one of the key results of this paper: For each possible arrival rate vector with mean $\lambda < S_{max}$ (so that it is strictly within the stability region of the system), if the system scale is large enough, this result shows that the SSG MaxWeight algorithm (that does not use downlink queue lengths) keeps the system stable. As we discussed earlier in Example 1, this is not true in general. The proof leverages the fact that the degrees of freedom resulting from the large number of channels compensates for any possible routing errors due to a lack of knowledge of downlink queues.

This result follows from channel diversity since under Assumption 2(a), the system is stable even if only a finite number of users have non-zero arrival rates.

As mentioned before, this result can be extended to $k-$hop networks. Please refer to Appendix C for the details.

Assumption 3: (HD-wDL: Stability)

- **Assumption 3(a)**:
  Arrivals and Bounded Channels
  - We assume that the arrival process is stationary, ergodic and i.i.d. across time-slots. We define $\lambda = \frac{1}{n}E\left[\sum_{i=1}^{n} A_i(0)\right]$. Then,
  $$P\left(\sum_{i=1}^{n} A_i(t) = n(\lambda+\delta)\right) \leq e^{-nk(\delta)}.$$
  - $A_i(t) \leq k_1 n^{\alpha}$ for some $\alpha < 1$, all t and i and some constant k_1.
  - The channel processes are i.i.d. across time-slots.
  - $X_{i,j}^{B,r}(t) \leq S_{max} < \infty$.
  - $X_{i,j}^{r}(t) \leq S_{max} < \infty$.
  - $X_{i,j}(t) \leq S_{max} < \infty$.
  - For every i, j, r and t,
  $$P(X_{i,j}(t) = S_{max}) = q(i,j,r) > 0.$$
- **Assumption 3(b)**:
  Let I be a set of users such that that $|I| \geq k_2 n$ for some $k_2 < 1$. Consider the event G that for a channel j and for every user $i \in I$, $X_{i,j}(t) < S_{max}$. Then,
  $$P(G) \leq o\left(\frac{1}{n^3}\right).$$
  If the channels are i.i.d. Bernoulli with parameter q across relays and time, we have that
  $$P(G) \leq (1-q)^{k_2 n}.$$
  This assumption, we can show, is also satisfied if the α mixing condition defined in [2] and discussed in Assumption 2(b) holds true for the channel variables.
- **Assumption 3(c)**:
  Let I be a set of users such that that $|I| = k_3 n$ and let J be a set of channels such that $|J| = \frac{2k_3 n}{q_{min}}$, where
  $$q_{min} = \min_{i,j,t} q(i,j,t) > 0.$$
  Consider the event W that for every user in I there exist $k_3 n$ channels in J such that $X_{i,j}(t) = S_{max}$. Then,
  $$P(W^c) = o\left(\frac{1}{n^2}\right).$$
  If the channels are i.i.d. Bernoulli with parameter q across relays and time, we have that
  $$P(W^c) = nk_3 e^{-\frac{2k_3 n}{q} H(\frac{q}{2}|q)}.$$
  This assumption, we can show, is also satisfied if the α mixing condition defined in [2] and discussed in Assumption 2(b) holds true for the channel variables.

Lemma 2. *Under Assumption 3, if* $\frac{1}{n}E\left[\sum_{i=1}^{n} A_i(0)\right] = \lambda > S_{max}$*, no scheduling algorithm can stabilize the system.*

Therefore, $\lambda \leq S_{max}$ is a necessary condition for an arrival vector to lie in the stability region of the system.

Theorem 3. *Under Assumption 3, for any arrival process with mean* $\lambda < S_{max}$*, the SSG MaxWeight algorithm stabilizes the HD-wDL system, i.e., the markov chain* $\{\boldsymbol{Q}(t), \boldsymbol{R}(t), \boldsymbol{A}(t)\}$ *is positive recurrent for* $n > n_0$ *where* n_0 *is a function of* λ.

Theorems 2 and 3 together form one of the two key messages of this paper which is that even though MaxWeight type algorithms are not throughput optimal for multihop networks in general, in the setting we consider i.e. large-scale multi-channel downlink networks with relays, they stabilize the system.

The proofs of Theorems 2 and 3 differ from the classical methods of proving stability because of the coupling between the base-station and relay queues. Please refer to Appendix A for the details of the proofs.

We note that the main difference between Assumptions 2 and 3 is that Assumption 2 (FD-w/oDL) is satisfied by all arrival processes such that the mean arrivals for each user is $\leq kn$ for any constant k (specifically, any $k < S_{max}$ works) whereas, Assumption 3 (HD-wDL) only allows arrival process which have mean $\leq k'n^{\alpha}$ for any constant k' and $\alpha < 1$. In particular, this implies that the SSG MaxWeight algorithm with Full Duplex relays can support any point that lie within the interior of the stability region, for n large enough[1]. This follows because the peak channel rate is S_{max}; thus, the maximum rate per user that can be supported by *any* algorithm is no more than $S_{max}n$.

On the other-hand, for the Half Duplex system with a direct link, Assumption 3 restricts the per-user arrival process (both mean and peak) to scale no more than $\leq k'n^{\alpha}$. This implies that in this setting, we can provably stabilize systems for which the arrival rates (across users) are more balanced, specifically, no single user can use the entire capacity.

B. Performance Analysis

Assumption 4: (FD-w/oDL: Performance Analysis)

- Bernoulli Arrivals and ON-OFF Channels
  - $A_i(t) = Bernoulli(p)$ i.i.d. across users and time-slots.
  - $X_{i,j}^{B,r}(t) = Bernoulli(q_2)$ i.i.d. across channels and time-slots.
  - $X_{i,j}^{r}(t) = Bernoulli(q_3)$ i.i.d. across channels and time-slots.
- Linearly Scaling Relays
  $$R(n) = \tilde{R}n, \tilde{R} > 0.$$

[1] Further, we can show that even with a Direct Link between the base-station and the Users, the analogous result goes through.

Our proofs work for any value of $\tilde{R}$, however we focus on the more realistic case of $\tilde{R} < 1$.

For the case of Bernoulli Arrivals and ON-OFF Channels, in addition to the BackPressure and SSG MaxWeight algorithms we also analyze two other algorithms derived from the Iterated Longest Queue First (ILQF) algorithm introduced in [3] which is known to be buffer-usage rate-function optimal for single hop networks (and thus is a good baseline for comparison).

This algorithm operates iteratively, where, in each iteration the algorithm determines a maximum size matching between the collection of longest queues and ON unallocated channels. After doing so, the queue lengths are updated, and the matching process repeats. The complete description of the algorithm is available in [3], Definition 3. We build on the ILQF algorithm to design scheduling and routing algorithms for multihop downlink networks.

1) ILQF BackPressure for FD-w/oDL: The allocation for relay queues is carried out first using the ILQF rule (tie breaking rule: highest priority is the smallest relay index followed by the smallest user index). The updated relay queue lengths are used for allocation of channels at the BS using the ILQF rule with the weight of each link being the backpressure of that link (tie breaking rule: highest priority is the smallest relay index followed by the smallest user index at each relay).

2) ILQF MaxWeight for FD-w/oDL: The allocation for relay queues is carried out first using the ILQF rule (tie breaking rule: highest priority is the smallest relay index followed by the smallest user index). The allocation for the BS queues is also done using the ILQF rule with the weight of each link being the queue length of that link (tie breaking rule: highest priority is the smallest relay index followed by the smallest user index at each relay).

We now analyze the performance of algorithms of the 4 algorithms for the FD-w/oDL system for the restricted class of arrival and channel processes characterized in Assumption 4. The performance metric we are interested in is the small buffer overflow probability which is the probability that the maximum queue length in the system (both at the base-station and the relays) is greater than a positive integer b. Formally, for each of these algorithms, we are interested in computing $c(b)$ where

$$c(b) = \frac{1}{b+1}\min\left\{\liminf_{n\to\infty}\frac{-1}{n}\log P\left(\max_{i,r} R_{ri}(0) > b\right),\right.$$
$$\left.\liminf_{n\to\infty}\frac{-1}{n}\log P\left(\max_{1\le i\le n} Q_i(0) > b\right)\right\},$$

for any fixed non-negative integer b.

Theorem 4. *Under Assumption 4, for the BackPressure algorithm,*

$$c(b)^{(BP)} = 0.$$

This theorem shows that even though the BackPressure algorithm is throughput optimal, it performs poorly when it comes to keeping the queue lengths small. This empirically holds even in the non-asymptotic region as seen in Figure 3.

Theorem 5. *Under Assumption 4, for the SSG MaxWeight algorithm, for any* $\epsilon \in (0, 1-p)$ *and*

$$\delta \in \left(0, \frac{q_3(1-p-\epsilon)}{2-q_3}\right),$$

$$c(b)^{(SMW)} \ge \min\left(H\big(p|p+\epsilon\big), \delta\log\frac{1}{1-q_3}, \frac{2\delta H\big(q_3|\frac{q_3}{2}\big)}{q_3}\right).$$

This is the second key result of this paper. This theorem shows that for the setting that we consider in Assumption 4, the SSG MaxWeight algorithm not only stabilizes the system for n large enough, but also performs well when it comes to keeping the queue lengths small.

Theorem 6. *Under Assumption 4, for the ILQF MaxWeight algorithm,*

$$c(b)^{(IMW)} \ge \min\left(\tilde{R}\log\frac{1}{1-q_2}, \frac{1}{2}\log\frac{1}{1-q_3}\right).$$

Theorem 7. *Under Assumption 4, for the ILQF BackPressure algorithm,*

$$c^{(IBP)}(b) \ge \min\left(\frac{1}{\lceil\frac{2}{\tilde{R}}\rceil}\log\frac{1}{1-q_2}, \frac{1}{\lceil\frac{2}{\tilde{R}}\rceil+1}\log\frac{1}{1-q_3}\right).$$

Since $\left\lceil\frac{2}{\tilde{R}}\right\rceil \ge \frac{2}{\tilde{R}} > \frac{1}{\tilde{R}}$ and $\left\lceil\frac{2}{\tilde{R}}\right\rceil + 1 \ge 2$ for all positive values of $\tilde{R}$, we observe from Theorems 6 and 7 that we get better bounds on the rate function for the ILQF MaxWeight algorithm than the ILQF BackPressure algorithm. The intuition for the improvement is clear: by not considering downlink backlogs, upstream nodes with the ILQF MaxWeight algorithm are more aggressive in using good channels to "push" packets closer toward the destination, and thus we expect, will result in a better performance than ILQF BackPressure. We further observe that the bound for the SSG MaxWeight algorithm in Theorem 5 is independent of $\tilde{R}$. Therefore for small enough values of $\tilde{R}$, i.e. for a small number of relays, we get better bounds on the performance of the SSG MaxWeight algorithm than the ILQF BackPressure algorithm. However, formally since these are bounds, we compare their relative delay performance through simulations in Section VI, which verify the intuition from the bounds.

To prove Theorems 5, 6 and 7 we use technical results on Markov Chain coupling from [5]; however, our algorithm performance analysis substantially differs from [5] as we need to deal with two hops (and can generalize to any finite number of hops), thus introducing coupled queues across hops. This entails a different proof technique.

The good performance of the iterative algorithms comes from the interplay between the large number of channels as well as users.

Please refer to Appendix B for the details.

VI. Simulation Results

We compare the end-to-end delay performance of four algorithms (BackPressure, SSG MaxWeight, ILQF MaxWeight and ILQF BackPressure) for a FD-w/oDL system. The end-to-end delay of a packet is defined as the number of time-slots it

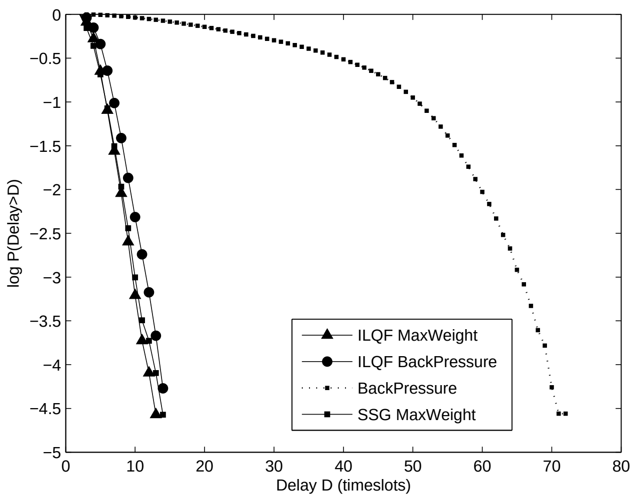

Fig. 3. *End-to-end delay performance of BackPressure, SSG MaxWeight, ILQF MaxWeight and ILQF BackPressure algorithms for a FD-w/oDL system consisting of 50 users and channels with 2 relays for load* $= 0.74$ *and ON-OFF channels with parameters 0.5 and 0.1 for the base-station to relay channels and relay to user channels respectively.*

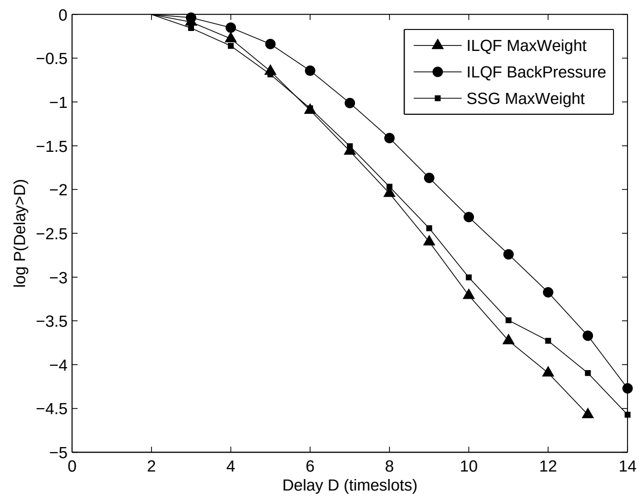

Fig. 4. *End-to-end delay performance of SSG MaxWeight, ILQF MaxWeight and ILQF BackPressure algorithms for a FD-w/oDL system consisting of 50 users and channels with 2 relays for load* $= 0.74$ *and ON-OFF channels with parameters 0.5 and 0.1 for the base-station to relay channels and relay to user channels respectively.*

spends in the system before reaching the intended user. This includes the time-slot at the beginning of which the packet arrives at the base-station. We consider end-to-end delay as the metric in the simulations because minimizing delay is important for several real-time applications (e.g., video, voice-over-IP). It is well known that delay is closely related to the queue-length at the base-station and the relays where the packets are temporarily stored on their way to the intended users. Therefore, we expect that algorithms which have good buffer-usage/queue-length performance, also have good end-to-end delay performance.

For this particular experiment, we assume that the system has 50 users, 50 channels and 2 relays. In addition, we assume that $p = 0.74$, $q_2 = 0.5$, $q_3 = 0.1$. We ran the system for 10000 time-slots. Figure 3 shows the delay performance of all 4 algorithms and Figure 4 is the same plot, but zoomed in to get a closer look at the difference in the performance of the three iterative algorithms. We see that the iterative algorithms perform much better than the non-iterative versions. The SSG MaxWeight algorithm seems to be doing better than ILQF BackPressure confirming our intuition that upstream nodes are more aggressive in the SSG MaxWeight algorithm because of the lack of downlink queue length information, leading to better delay performance. This result also validates the difference in the bounds obtained in Theorems 5, 6 and 7.

VII. Conclusions

We proved that variants of the MaxWeight algorithm are stabilizing for large scale relay networks under appropriate models. We compared the performance of Iterative MaxWeight algorithms and Iterative BackPressure algorithm and found that the Iterative MaxWeight algorithms have better performance. Given that the complexity of these algorithms are not significant (low-degree polynomial, please see [4] for discussion on the complexity of SSG-like algorithms), they can be considered for implementation in practical settings.

References

[1] 3gpp tr 25.913. requirements for evolved utra (e-utra) and evolved utran (e-utran). March, 2006.

[2] P. Billingsley. *Probability and Measure*. Wiley,, 1995.

[3] S. Bodas, S. Shakkottai, L. Ying, and R. Srikant. Scheduling in multichannel wireless netowrks: Rate function optimality in the small buffer regime. In *Proceedings of SIGMETRICS/performance Conf.*, 2009.

[4] S. Bodas, S. Shakkottai, L. Ying, and R. Srikant. Low-complexity scheduling algorithms for multi-channel downlink wireless networks. In *Proceedings of IEEE Infocom*, 2010.

[5] S. Bodas, S. Shakkottai, L. Ying, and R. Srikant. Scheduling for small delay in multi-rate multi-channel wireless networks. In *Proceedings of IEEE Infocom*, 2011.

[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 Trans. Network.*, 19(6):1597–1609, 2011.

[7] A. Eryilmaz, R. Srikant, and J. Perkins. Stable scheduling policies for fading wireless channels. *IEEE/ACM Trans. Network.*, 13:411–424, April 2005.

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

[9] B. Ji, C. Joo, and N. Shroff. Throughput-optimal scheduling in multi-hop wireless networks without per-flow information. In *Proceedings of WiOPT*, 2011.

[10] Bo Ji, Gagan R Gupta, Manu Sharma, Xiaojun Lin, and Ness B Shroff. Achieving optimal throughput and near-optimal asymptotic delay performance in multi-channel wireless networks with low complexity: A practical greedy scheduling policy. *arXiv preprint arXiv:1212.1638*, 2012.

[11] S. Liu, E. Ekici, and L. Ying. Scheduling in multihop wireless networks without back-pressure. In *Annual Conference on Communication, Control and Computing (Allerton)*, 2010.

[12] http://www.3gpp.org/lte-advanced.

[13] S. Moharir and S. Shakkottai. Maxweight vs backpressure: Routing and scheduling for multi-channel relay networks. In *Proceedings of IEEE Infocom*, Turin, Italy, April 2013.

[14] S. Moharir and S. Shakkottai. Maxweight vs backpressure: Routing and scheduling for multi-channel relay networks. Technical report, 2014.

[15] M. Neely, E. Modiano, and C. Rohrs. Dynamic power allocation and routing for time-varying wireless networks. *IEEE J. Sel. Areas Commun.*, 23(1):89–103, 2005.

[16] S. Shakkottai. Effective capacity and qos for wireless scheduling. *IEEE Trans. Automat. Contr.*, 53(3):749–761, 2008.

[17] M. Sharma and X. Lin. Ofdm downlink scheduling for delay-optimality: Many-channel many-source asymptotics with general arrival processes. In *Proceedings of ITA*, 2011.

[18] A. Stolyar. Large deviations of queues sharing a randomly time-varying server. *Queueing Systems*, 59(2):1–35, 2008.

[19] A. Stolyar. Large number of queues in tandem: Scaling properties under back-pressure algorithm. *Queueing Systems*, 67(2):111–126, 2011.

[20] L. Tassiulas and A.Ephermides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Trans. Automat. Contr.*, 37(12):1936–1948, 1992.

[21] L. Tassiulas and A.Ephermides. Dynamic server allocation to parallel queues with randomly varying connectivity. *IEEE Trans. Automat. Contr.*, 39:466–478, 1993.

[22] V. Venkataramanan and X. Lin. Structural properties of ldp for queue-length based wireless scheduling algorithms. In *Annual Conference on Communication, Control and Computing (Allerton)*, 2007.

[23] V. Venkatramanan, X. Lin, L. Ying, and S. Shakkottai. On scheduling for minimizing end-to-end buffer usage over multihop wireless networks. In *Proceedings of IEEE Infocom*, 2010.

[24] L. Ying, S. Shakkottai, and A. Reddy. On combining shortest-path and back-pressure routing over multihop wireless networks. In *Proceedings of IEEE Infocom*, 2009.

[25] L. Ying, R. Srikant, A. Eryilmax, and G. Dullrud. A large deviations analysis of scheduling in wireless networks. *IEEE Trans. Inform. Theory*, 52(11):5088–5098, 2006.

Sharayu Moharir is a Ph.D. student at the Department of Electrical and Computer Engineering at the University of Texas at Austin. She received her M.Tech. in Comminication and Signal Processing and her B.Tech. in Electrical Engineering from the Indian Institute of Technology, Bombay in 2009.

Her research interests include algorithms and performance analysis for wireless networks and content delivery networks.

Sanjay Shakkottai (M'02–SM'11–F'14) Sanjay Shakkottai received his Ph.D. from the ECE Department at the University of Illinois at Urbana-Champaign in 2002. He is with The University of Texas at Austin, where he is currently a Professor in the Department of Electrical and Computer Engineering. He received the NSF CAREER award in 2004, and was elected as an IEEE Fellow in 2014. His current research interests include network architectures, algorithms and performance analysis for wireless networks, and learning and inference over social networks.

Appendix A
Large System Stability of Iterative Max Weight

We consider the FD-w/oDL and HD-wDL models described in Section II separately. We first provide a proof outline for the FD-w/oDL model.

A. FD-w/oDL

1) We first prove that if $\lambda > S_{max}$, no scheduling policy can stabilize the system.
2) We then show that the base-station queues are stable for any $\lambda < S_{max}$. This proof uses the fact that since there are $R(n)$ relays, for large n, every channel can be used at S_{max} to send packets to at least one of these relays with very high probability. As $\lambda < S_{max}$, with high probability, fewer packets come into the system in a time slot than the number that can be served, thus ensuring that the base-station queues are stable.
3) Since the arrival process at the base-station queues is stationary and ergodic, and the base-station queues are stable, the arrival process at the relay queues (which is the departure process of the base-station queues) is also stationary and ergodic. By Theorem 5 in [4], we know that the SSG algorithm is throughput optimal for the system consisting of just the relay queues. Therefore, to prove that the MaxWeight SSG algorithm stabilizes the relay queues, we need to show that the arrival process at the relays is inside the throughput region of the relays queues.
4) Since the throughput region of the relays queues is not known, to do this, we propose an algorithm called the Arrival Prioritized-SSG (AP-SSG) algorithm and show that this algorithm can stabilize the relay queues for the arrival process which is the departure process of the base-station queues. This shows that the departure process of the base-station queues lies in the throughput region of the relay queues and therefore, the relay queues will also be stabilized by the throughput optimal SSG algorithm.
5) The AP-SSG algorithm stores 2 values corresponding to each relay queue. Before allocation for slot t begins, the first value $A_i^{r(0)}$ is initialized to the number of arrivals to that queue at the beginning of slot t and the second value $R_{ri}^{(0)}$ is initialized to the queue length of the queue for user i at relay r at the end of time-slot $t-1$.
The allocation proceeds in n rounds. In round k, the algorithm finds a queue with the highest $A_i^{r(k-1)} X_{i,k}^r$ value. If this value is greater than 0, channel k is allocated to queue i at relay r and $A_i^{r(k)}$ is updated to $(A_i^{r(k-1)} - X_{i,k}^r)^+$. If $A_i^{r(k-1)} X_{i,k}^r = 0$, the algorithm finds a queue with the highest $R_{ri}^{(k-1)} X_{i,k}^r$ value and serves it. It updates $R_{ri}^{(k)}$ to $(R_{ri}^{(k-1)} - X_{i,k}^r)^+$.
The AP-SSG algorithm therefore, gives the first priority to queues which have packets that arrived at the beginning of that slot and then to queues which are the most backlogged. For the AP-SSG algorithm, we prove the following key lemma.

Lemma 3. *Let S_{ri} be the service allocated to queue i at relay r by the AP-SSG algorithm. Let E_4 be the event that*

$$\cap_{r,i}\{A_i^r \leq S_{ri}\} \cap \{S_{r^*i^*} \geq A_{i^*}^{r^*} + S_{max}\},$$

where $\{r^*, i^*\} \in \arg\max_{r,i} R_{ri}(t-1)$. The event E_4 implies that all the arrivals to the relay queues at the *beginning of slot t are served in slot t and the at least one of the longest relay queues is served by at least 1 additional channel at S_{max}. Then, under Assumption 2,*

$$P(E_4^c) = o\bigg(\frac{1}{n}\bigg).$$

The above lemma essentially shows a negative drift of at least $R_{max}S_{max}$, where R_{max} is the maximum queue length of the relay queues at the end of time-slot $t-1$. We then show that there exists an n_0 such that this algorithm stabilizes the relay queue system with $n > n_0$ channels via the quadratic lyapunov function. This proves that the arrival process at the relay queues which is the departure process of the base-station queues lies inside the throughput region of the relay queues and therefore, the relay queues will be stabilized by the SSG algorithm.

The following Lemma generalizes Theorem 4 in [5]. Theorem 4 in [5] was restricted to computing the stationary distribution of Markov Chains such that in each time-slot, the value of the Markov random variable could increase by at most a constant number (k_0) with exponentially small probability (e^{-cn}). This lemma generalizes the theorem to markov chains which increase by at most $\chi(n)$ in a given slot with probability at most $f(n)$ such that $\chi(n)^3 f(n) = o(1/n^2)$.

Lemma 4. *Consider a discrete time Markov Chain $Y^{(n)} \in \{0, 1, 2, ...\}$. Let $f(n) = o\big(\frac{1}{n^6}\big)$ and $\chi(n)$ such that $\chi(n)^3 f(n) = o(1/n^2)$. Suppose that the transition probabilities are as follows:*

If $Y^{(n)}(t) > 0$,

$$\begin{aligned} P(Y^{(n)}(t+1) = Y^{(n)}(t) - 1) &= 1/2 \\ P(Y^{(n)}(t+1) = Y^{(n)}(t) + \chi(n)) &= f(n) \\ P(Y^{(n)}(t+1) = Y^{(n)}(t)) &= 1/2 - f(n). \end{aligned}$$

If $Y^{(n)}(t) = 0$,

$$\begin{aligned} P(Y^{(n)}(t+1) = \chi(n)) &= f(n) \\ P(Y^{(n)}(t+1) = 0) &= 1 - f(n). \end{aligned}$$

Let $\pi(m) = P(Y^{(n)}(t) = m)$. For this Markov Chain, we have that,

$$1 - \pi(0) \leq 4\chi(n)^3 f(n) = o\bigg(\frac{1}{n^2}\bigg).$$

Proof: Consider the Lyapunov function *Lyap(x)=x*. For n large enough, we have

$$\begin{aligned} E(Y^{(n)}(t+1) &- Y^{(n)}(t)|Y^{(n)}(t), Y^{(n)}(t) > 0) \\ &\leq \chi(n)f(n) - \frac{1}{2} \\ &\leq -\frac{1}{3}, \end{aligned}$$

so the Lyapunov function has negative drift outside the set $\{0\}$ and therefore the Markov Chain is positive recurrent. The Markov Chain is also irreducible and aperiodic and therefore has a unique stationary distribution. We prove the following statement by induction about $\pi(m)$ by induction

$$\pi(m) \leq \pi(0)(2\chi(n))^{2\lceil m/\chi(n) \rceil} f(n)^{\lceil m/\chi(n) \rceil}.$$

For n large enough, $2\chi(n)^2 f(n) < 1$.

Case I: $1 \leq m \leq \chi(n)$

$$\begin{aligned} \pi(m) &= 2\sum_{r=1}^{m} \pi(m-r) \sum_{j=r}^{m} f(n) \\ &\leq 2m^2 \pi(0) f(n) \\ &\leq 2(\chi(n))^2 \pi(0) f(n). \end{aligned}$$

Case II: $(k-1)\chi(n) < m \leq k\chi(n)$

$$\begin{aligned} \pi(m) &= 2\sum_{r=1}^{\chi(n)} \pi(m-r) \sum_{j=r}^{\chi(n)} f(n) \\ &\leq 2(\chi(n))^2 \pi(m - \chi(n)) f(n) \\ &\leq 2\chi(n)^2 \pi(0) 2^{k-1} (\chi(n))^{k-1} f(n)^{k-1} f(n) \\ &= (2\chi(n))^{2k} \pi(0) f(n)^k, \end{aligned}$$

thus completing the proof by induction.

Let n be large enough such that $W = 2\chi(n)^3 f(n) < 1/2$, then, by adding the values of $\pi(m)$ for $m = 0$ to ∞ and equating it to 1, we get that,

$$\begin{aligned} 1 - \pi(0) &\leq \frac{W}{1-W} \\ &\leq 2W \\ &= 4\chi(n)^3 f(n). \end{aligned}$$

■

In the following lemma, we prove that if on average, more than nS_{max} packets come into the system in every slot, no scheduling policy can stabilize the system.

Lemma (1). *Under Assumption 2, if $\frac{1}{n}E\left[\sum_{i=1}^{n} A_i(0)\right] = \lambda > S_{max}$, then the system is unstable under any scheduling algorithm.*

Proof: If $\lambda > S_{max}$, then the mean number packet arrivals to the system in a given time-slot is more than the maximum number of packets that can be served by the base-station or the relays in a given time-slot ($= nS_{max}$). Hence the system is unstable under any scheduling algorithm.

■

We now prove that if $\lambda < S_{max}$, the SSG MaxWeight algorithm stabilizes the system. To handle coupled queues across hops (and the routing induced by muliple hops and paths), our proof is iterative across hops. We first look at the base-station queues.

Lemma 5. *Under Assumptions 2 and the SSG MaxWeight algorithm, given any arrival process such that $\lambda < S_{max}$, the markov chain $(\boldsymbol{Q}(t), \boldsymbol{A}(t))$ is positive recurrent for n large enough.*

Proof: We say that the base-station queue are stable if the SSG MaxWeight algorithm makes the base-station queues an aperiodic Markov Chain with a single communicating class which is positive recurrent.

Consider the Markov chain $(\mathbf{Q}(t), \mathbf{A}(t))$ and the lyapunov function $Q(t)$ where $Q(t) = \sum_{i=1}^{n} Q_i(t)$.
Consider the finite set $F = \{\mathbf{Q} : \max_{1 \le i \le n} Q_i \le nS_{max}\}$. In this set,

$$\begin{aligned} & E[Q(t+1) - Q(t)|\mathbf{Q}(t), \mathbf{A}(t)] \\ = \;& E\Big[\sum_{i=1}^{n} Q_i(t+1) - \sum_{i=1}^{n} Q_i(t)\Big|\mathbf{Q}(t), \mathbf{A}(t)\Big] \\ \le \;& n\lambda < \infty, \end{aligned}$$

by Assumption 2(a). Outside the set F,

$$\begin{aligned} & E[Q(t+1) - Q(t)|\mathbf{Q}(t), \mathbf{A}(t)] \\ = \;& E\Big[\sum_{i=1}^{n} Q_i(t+1) - \sum_{i=1}^{n} Q_i(t)\Big|\mathbf{Q}(t), \mathbf{A}(t)\Big] \\ = \;& E\Big[\sum_{i=1}^{n}\Big(Q_i(t) + A_i(t+1) - \\ & \sum_{j=1}^{n} X_{i,j}^{B,r}(t+1)Y_{i,j}^{B,r}(t+1)\Big)^{+} - Q(t)\Big|\mathbf{Q}(t), \mathbf{A}(t)\Big] \\ \stackrel{(a)}{=} \;& E\Big[\sum_{i=1}^{n} A_i(t+1) \\ & -\sum_{j=1}^{n} X_{i,j}^{B,r}(t+1)Y_{i,j}^{B,r}(t+1)\Big|\mathbf{Q}(t), \mathbf{A}(t)\Big] \\ = \;& E\Big[\sum_{i=1}^{n} A_i(t+1)\Big|\mathbf{Q}(t), \mathbf{A}(t)\Big] \\ & -E\Big[\sum_{j=1}^{n} X_{i,j}^{B,r}(t+1)Y_{i,j}^{B,r}(t+1)\Big|\mathbf{Q}(t), \mathbf{A}(t)\Big] \\ = \;& n\lambda - E\Big[\sum_{j=1}^{n} X_{i,j}^{B,r}(t+1)Y_{i,j}^{B,r}(t+1)\Big|\mathbf{Q}(t), \mathbf{A}(t)\Big]. \end{aligned}$$

Where (a) follows from the fact that outside the set F, since $\max_{1 \le i \le n} Q_i > nS_{max}$ the base station always has packets to send on all channels, therefore, no capacity is wasted. Let $3\epsilon = \frac{S_{max} - \lambda}{S_{max}}$. Consider the event E that there exists a set J of channels such that $|J| = 2n\epsilon$ and $X_{i,j}^{B,r} < S_{max}$ for all $j \in J$ and $1 \le r \le R(n)$.

$$\begin{aligned} & E\Big[\sum_{j=1}^{n} X_{i,j}^{B,r}(t+1)Y_{i,j}^{B,r}(t+1)\Big|E^c\Big] \ge (1-2\epsilon)S_{max}n, \\ & E\Big[\sum_{j=1}^{n} X_{i,j}^{B,r}(t+1)Y_{i,j}^{B,r}(t+1)\Big|E\Big] \ge 0. \end{aligned}$$

Therefore,

$$\begin{aligned} & E\Big[\sum_{j=1}^{n} X_{i,j}^{B,r}(t+1)Y_{i,j}^{B,r}(t+1)\Big] \\ = \;& E\Big[\sum_{j=1}^{n} X_{i,j}^{B,r}(t+1)Y_{i,j}^{B,r}(t+1)\Big|E\Big]P(E) \\ & +E\Big[\sum_{j=1}^{n} X_{i,j}^{B,r}(t+1)Y_{i,j}^{B,r}(t+1)\Big|E^c\Big]P(E^c) \\ \ge \;& (1-2\epsilon)S_{max}nP(E^c). \end{aligned}$$

By Assumption 2(b), $P(E^c) = o\Big(\frac{1}{n^6}\Big)$ and therefore, for $\lambda < S_{max}$ and n large enough,

$$\begin{aligned} & E[Q(t+1) - Q(t)|\mathbf{Q}(t), \mathbf{A}(t)] \\ \le \;& n\lambda - (1-2\epsilon)S_{max}nP(E^c) \\ \le \;& -1/2. \end{aligned}$$

Therefore, by Foster's theorem, the Markov Chain $\mathbf{Q}(t)$ is positive recurrent.
Now consider the Markov Chain $Q(t)$. We need to compute $P(Q(t) > 0)$ to prove that the relay queues are stable. To this end, we study the Markov Chain $Y^{(n)}(t)$ defined in Lemma 4 for $f(n) = o(1/n^6)$ and $\chi(n) = k_1 n^2$. Note that by Theorem 3 in [5], $Q(t) \le_{st} Y^{(n)}(t)$ where $Q(t) \le_{st} Y^{(n)}(t) \Rightarrow P(Q(t) > x) \le P(Y^{(n)}(t) > x)$, $\forall x$. By Lemma 4 we have that, for n large enough, for the Markov Chain $Y^{(n)}(t)$,

$$\begin{aligned} 1 - \pi(0) \;&\le\; \frac{W}{1-W} \\ &\le\; 2W \\ &=\; 4(k_1 n^2)^2 P(E^c). \end{aligned}$$

Therefore, $P(Q(t) > 0) \le 4{k_1}^2 n^4 P(E^c) = o\Big(\frac{1}{n^2}\Big)$.

■

We now look at the relay queues. We note that the departure process of the base-station queues is the arrival process of the relay queues. Since the arrival process of the base-station queues is stationary and ergodic and the base-station queue system is stable, the departure process is also stationary and ergodic and therefore, the arrival process of the relay queues is stationary and ergodic. Additionally, if we prove that the departure process of the base-station queues is inside the throughput region of the relay queues, then we have that the SSG algorithm will stabilize the relay queues. Since the SSG algorithm is throughput optimal for the system consisting of just the relay queues and users by Theorem 5 in [4].

To prove that the departure process of the base-station queues is inside the throughput region of the relay queues, we prove that there exists an algorithm that can stabilize the relay queues for the arrival process which is the departure process of the base-station queues. We call this algorithm the Arrival Prioritized-SSG (AP-SSG) algorithm.
Definition: The AP-SSG algorithm allocates channels to queues in time-slot t according to the following procedure.
Input:

1) The queue lengths $R_{ri}(t-1)$, for $1 \leq i \leq n$, $1 \leq r \leq R(n)$.
2) The arrival vector $A_i^r(t)$, for for $1 \leq i \leq n$, $1 \leq r \leq R(n)$.
3) The channel realizations $X_{ij}^r(t)$, for $1 \leq i \leq n$, $1 \leq r \leq R(n)$, $1 \leq j \leq n$.

Steps:

1) Initialize $k = 1$ and $Y_{ij}^r(t) = 0$, $R_{ri}^{(0)}(t) = R_{ri}(t)$, $A_i^{r(0)}(t) = A_i^r(t)$ for $1 \leq i \leq n$, $1 \leq r \leq R(n)$, $1 \leq j \leq n$.
2) In the k^{th} round of allocation, search for the relay and queue index
$$\{r^*, i^*\} \in \arg \max_{1\leq i\leq n, 1\leq r\leq R(n)} A_i^{r(k-1)} X_{ij}^r(t),$$
breaking ties in the favor of the smaller relay index, followed by the smaller queue index. If $A_{i^*}^{r(k-1)} X_{i^*j}^{r^*}(t) > 0$, goto step 3. Else goto step 4.
3) Allocate channel k to serve $R_{r^*i^*}$, define $Y_{i^*k}^{r^*}(t) = 1$ and update the value of $A_{i^*}^{r^*(k)}$ to $(A_{i^*}^{r^*(k-1)} - X_{i^*j}^{r^*}(t))^+$. Goto Step 5.
4) Search for the relay and queue index
$$\{r^*, i^*\} \in \arg \max_{1\leq i\leq n, 1\leq r\leq R(n)} R_{ri}^{(k-1)} X_{ij}^r(t),$$
breaking ties in the favor of the smaller relay index, followed by the smaller queue index. Allocate channel k to serve $R_{r^*i^*}$, define $Y_{i^*k}^{r^*}(t) = 1$ and update the value of $R_{r^*i^*}^{(k)}$ to $(R_{r^*i^*}^{(k-1)} - X_{i^*j}^{r^*}(t))^+$.
5) If $k = n$, stop, else increment k by 1 and goto step 2.

We now define a series of events and compute their probabilities.

Lemma 6. *Under Assumption 2 and the SSG MaxWeight algorithm, let E_0 be the event that the max queue-length of the base-station queues at the end of slot t is 0. Then,*
$$P(E_0^c) = o\left(\frac{1}{n^3}\right).$$

Proof: Follows by Lemma 5. ■

Lemma 7. *Let $3\epsilon = S_{max} - \lambda$. Under Assumption 2 and the SSG MaxWeight algorithm, let E_1 be the event that the max arrivals to the base-station queues at the beginning of slot t is less than $n(\lambda + \epsilon)$. Then,*
$$P(E_1^c) = o\left(\frac{1}{n^3}\right).$$

Proof: Follows by Assumption 2(a). ■

Lemma 8. *Under Assumption 2(c) and the SSG MaxWeight algorithm, let E_2 be the event that the max arrivals to any relay queue in a given time-slot is less than $\frac{2nS_{max}}{R(n)}$. Then,*
$$P(E_2^c) = o\left(\frac{1}{n^3}\right).$$

Proof: Recall the tie-breaking policy of the SSG MaxWeight rule: initialize the priority order of the relays as $\{1, 2, ..., R(n)\}$. In each round of the allocation process, the relay that is allocated that particular channel is then removed from its current position in the priority order and inserted at the last position to get the new priority order. Consider a particular relay r which is allocated the j^{th} channel. It is then pushed to the end of the priority order. In the subsequent rounds of channel allocation, another channel will be allocated to it only if that channel cannot be used at S_{max} to send packets to any of the other relays which are higher than r in the priority list. Consider the next $R(n)/2$ rounds of channel allocation. In each of these rounds, there are at least $R(n)/2$ relays that have higher priority than relay r. Then, by Assumption 2(c) for $\delta = 0.5$, we have that the probability that relay r is allocated another channel in the next $R(n)/2$ rounds of channel allocation is $o(n^{-4})$. The result then follows from the union bound over all channels. ■

Let $E_3 = E_0 \cap E_1 \cap E_2$. By Lemma 6, 7 and 8, $P(E_3^c) = o(\frac{1}{n^3})$. In the following lemma, we prove that the AP-SSG algorithm stabilizes the relay queues. Then, using the fact that the SSG algorithm is throughput optimal for one hop networks, we prove that the SSG MaxWeight algorithm will also stabilize the relay queues.

Lemma (3). *Let $S_{ri} = \sum_{j=1}^n X_{ij}^r Y_{ij}^r$ be the service allocated to queue i at relay r by the AP-SSG algorithm. Under Assumption 2(c) and 2(d), let E_4 be the event that*
$$\cap_{r,i}\{A_i^r \leq S_{ri}\} \cap \{S_{r^*i^*} \geq A_{i^*}^{r^*} + S_{max}\},$$
where $\{r^*, i^*\} \in \arg\max_{r,i} R_{ri}(t-1)$. The event E_4 means that all the arrivals to the relay queues at the beginning of slot t are served in slot t and at least one of the longest relay queues is served by at least 1 additional channel. Then,
$$P(E_4^c) = o\left(\frac{1}{n}\right).$$

Proof: We condition the proof on E_3. Pick any δ in
$$\left(0, \frac{q_{min}(1 - \lambda - 2\epsilon)}{2S_{max}(2 - q_{min})}\right).$$

Let F_m be the set of relay queues which received m packets at the beginning of slot t. Conditioned on E_3, $|F_m| = 0$ for $m > \frac{2nS_{max}}{R(n)}$. Recall that $3\epsilon = S_{max} - \lambda$. Let $m = \frac{2nS_{max}}{R(n)}$.

Case I: $|F_m| = |F_m^{(0)}| \geq \delta R(n)$.

Define $w_0 = |F_m^{(0)}| - \delta R(n)$. By Assumption 2(c), we have that after the first w_0 rounds of service, $|F_m^{(w_0)}| \leq \delta R(n)$ w.p. $\geq 1 - \delta R(n) o(1/n^3)$.

Consider the next $v_0 = \dfrac{2\delta R(n)}{q_{min}}$ rounds of allocation, By Assumption 2(d), we have that $|F_m^{(v_0+w_0)}| = 0$ w.p. $\geq 1 - o(1/n^3)$.

Case II: $|F_m| = |F_m^{(0)}| \leq \delta R(n)$.

Consider the first $v_0 = \dfrac{2\delta R(n)}{q_{min}}$ rounds of allocation, By

Assumption 2(d), we have that $|F_m^{(v_0)}| = 0$ w.p. $\geq 1 - o(1/n^3)$.

The proof now follows by repeatedly applying the above procedure for $m = \frac{2nS_{max}}{R(n)} - 1, \frac{2nS_{max}}{R(n)} - 2, ...1$. As a result, all the new packets are served at the end of

$$\begin{aligned} & n(\lambda + \epsilon) - 2nS_{max}\delta\left(\frac{2}{q_{min}} - 1\right) \\ < \quad & n(1-\epsilon) \end{aligned}$$

rounds of allocation with probability

$$\geq 1 - P(E_3^c) + \frac{2n^2 S_{max}}{R(n)}\left(\delta R(n) o\left(\frac{1}{n^3}\right) + o\left(\frac{1}{n^3}\right)\right).$$

In the remaining ϵn rounds of allocation, by Assumption 2(d), at least one channel serves the longest relay queue with probability $= o(1/n^3)$. Therefore,

$$P(E_4^c) = o(1/n).$$

■

Lemma 9. *Under Assumptions 2 and the Iterative MaxWeight algorithm, for any arrival process with $\lambda < S_{max}$, the relay queues are stable for n large enough.*

Proof: Let $R(t+1) = R(t) + A(t) - S(t) + U(t)$ where $A(t), S(t)$ and $U(t)$ represent the arrivals, service and unused service respectively. Consider the Lyapunov function $V(t)$ where $V(\mathbf{R}(t)) = ||R(t)||^2$. We drop the time index for convenience.

$$\begin{aligned} & E[V(t+1) - V(t)|\mathbf{R}(t)] \\ = \quad & ||R(t+1)||^2 - ||R(t)||^2 \\ = \quad & ||R + A - S + U||^2 - ||R||^2 \\ = \quad & ||R||^2 + ||(A-S)||^2 + 2R(A-S) + ||U||^2 \\ & + 2\langle U, (R + A - S)\rangle - ||R||^2 \\ \leq \quad & n^2 S_{max}^2 + 2\langle R, (A-S)\rangle. \end{aligned}$$

We use the fact that $U = -(R + A - S)$, therefore $\langle U, (R + A - S)\rangle = -||U||^2 \leq 0$.
For the AP-SSG algorithm and the event E_4 defined above, $P(E_4^c) = o(1/n)$. By the definition of event E_4, we have that

$$E[\langle R, A - S\rangle|\mathbf{R}(t), E_4] \leq -R_{max}S_{max}.$$

Also,

$$E[\langle R, A - S\rangle|\mathbf{R}(t), E_4^c] \leq R_{max}S_{max}n.$$

Therefore,

$$\begin{aligned} & E[V(t+1) - V(t)|\mathbf{R}(t)] \\ \leq \quad & n^2 S_{max}^2 + 2\langle R, (A-S)\rangle. \\ \leq \quad & n^2 S_{max}^2 - 2R_{max}S_{max}P(E_4) + 2R_{max}S_{max}nP(E_4^c) \\ \leq \quad & n^2 S_{max}^2 - R_{max}S_{max}P(E_4), \end{aligned}$$

for n large enough. For $R_{max} > \frac{n^2 S_{max}^2 - 1/2}{P(E_4)S_{max}}$, the drift is $\leq -\frac{1}{2}$. Therefore, by Foster's theorem, the relay queues are stabilized by the AP-SSG algorithm. Further, by Theorem 5 in [4], the SSG algorithm is throughput optimal for a system consisting of just the relay queues. Since there exists an algorithm (AP-SSG) which can stabilize the relay queues, the SSG algorithm will also stabilize the relay queues.

■

Theorem (2). *Under Assumption 2, for arrival processes with $\lambda < S_{max}$, the SSG MaxWeight algorithm stabilizes the FD-w/oDL system, i.e., the markov chain $\{\boldsymbol{Q}(t), \boldsymbol{R}(t), \boldsymbol{A}(t)\}$ is positive recurrent for $n > n_0$ where n_0 is a function of λ.*

Proof: The proof follows from Lemma 5 and Lemma 9.

■

B. HD-wDL

This proof proceeds in the following three steps. Please refer to [14] for the complete proof.

1) We first prove that under Assumption 3, there are no arrivals to the relays at the beginning of a slot with probability $= o(1/n^2)$.
2) We then prove that with high probability, the maximum queue-length in the system does not increase in any time-slot.
3) Next, we prove that there exists a constant k_0 such that in k_0 consecutive time-slots, the maximum queue-length in the system decreases by 1 with probability $\geq 1/2$. We use the proof technique used in Lemma 8 in [4] to get this result.
4) Finally, We prove the stability of the system by constructing a Markov Chain of the maximum queue-length of the system. We then use Theorems 2 and 3 from [5] and Lemma 4 to prove stability of this Markov Chain, thus proving the stability of the HD-wDL system.

Appendix B
Performance Analysis

In this section, we analyze the rate function for the small buffer overflow probability of the BackPressure algorithm, the SSG MaxWeight algorithm, the ILQF MaxWeight algorithm and the ILQF BackPressure algorithm for the FD-W/oDL model.

A. BackPressure

We first show that the BackPressure algorithm has a zero rate for the small buffer overflow event. The proof follows on the same lines as the proof of Theorem 3 in [4]. In [4], it was proved that the maximum queue-length increases with at least a constant probability in each slot. We prove the same result for the backpressure value of the base-station queues and use the backpressure values as a lower bound for the queue-lengths to prove the desired result. Please refer to [14] for the complete proof.

B. SSG MaxWeight

The proofs for performance of the ILQF BackPressure algorithm, the ILQF MaxWeight algorithm and the SSG MaxWeight algorithm for the FD-w/oDL system proposed in

Section IV work in a sequential manner. We divide the set of queues into two sets: the base-station queues and the relay queues. Even though the two sets of queues are coupled, surprisingly, they can be analyzed in a sequential manner to provide performance guarantees on all the queue-lengths in the system.

For the ILQF BackPressure algorithm, we analyze the relay queues first and prove that they are all empty with probability $\approx e^{-nc}$ for some $c > 0$. We observe that at the base-station, the ILQF backpressure algorithm tries to serve queues with the highest backpressure values which are not always queues with maximum queue lengths. However, if the relay queues are all empty, the two sets are the same. We use this fact to analyze the maximum base-station queue length.

For the ILQF MaxWeight algorithm and the SSG MaxWeight algorithm, we analyze the base-station queues first and use that result to analyze the relay queues.

The analysis for each set of queues is carried out in the following steps:

1) We first show that for the set of queues that we are analyzing (either the relay queues or the base-station queues), the maximum queue length increases in a slot with a very small probability ($\leq e^{-nc}$).
2) Using Step 1 and Lemma 8 in [4], we show that there exists a constant k_0 such that in k_0 consecutive time-slots, with probability at least 1/2, the maximum queue length decreases by 1.
3) To compute the stationary distribution of the maximum value of queues in this set, we construct a Markov Chain $Y^{(n)}(t)$ which has the following properties:

$$\begin{aligned} P(Y^{(n)}(t+1) = (Y^{(n)}(t)-1)^+) &= 1/2 \\ P(Y^{(n)}(t+1) = Y^{(n)}(t)+\chi(n)) &= e^{-nc} \\ P(Y^{(n)}(t+1) = Y^{(n)}(t)) &= 1/2 - e^{-nc}. \end{aligned}$$

For the relay queues, $\chi(n) = k_0 n$. We prove that for $f(n) = e^{-nc}$ for some $c > 0$, we have that,

$$\liminf_{n\to\infty} \frac{-1}{n} \log P\Big(Y^{(n)}(0) > b\Big) \geq (b+1)c.$$

For the base-station queues, $\chi(n) = k_0$. Using Theorem 4 in [5], we have that,

$$\liminf_{n\to\infty} \frac{-1}{n} \log P\Big(Y^{(n)}(0) > b\Big) \geq (b+1)c.$$

4) We use Theorem 3 in [5] to prove that the maximum queue length in the set of interest is stochastically dominated by the process $Y^{(n)}(t)$ for the corresponding value of $\chi(n)$. We then use the stationary distribution of $Y^{(n)}(t)$ to get the desired result.

For the SSG MaxWeight algorithm, we first focus on the base-station queues and find the probability that in the steady state, the maximum queue-length is greater than b at the beginning of a slot. Conditioned on the fact that the longest base-station queue has b packets, at the end of time-slot $t-1$, not more than $b+1$ packet can arrive to any particular relay queue at the beginning of slot $t+1$. Using this, we find the probability that in the steady state, all relay queues have less than b packets at the end of a time-slot for all integers $b \geq 0$.

Basestation Queues

Lemma 10. *Fix a value of $\epsilon \in (0, 1-p)$. Define*

$$\xi_B(t) =: \max_{1\leq i\leq n} Q_i(t).$$

Then,

$$P(\xi_B(t) > \xi_B(t-1)) \leq e^{-c_B n^2 + k(\epsilon)n} + e^{-nH(p|p+\epsilon)}.$$

Proof: Consider the event E that

$$\sum_{i=1}^{n} A_i(t) \leq (p+\epsilon)n.$$

Then,

$$P(E^c) \leq e^{-nH(p|p+\epsilon)}.$$

We condition the rest of the proof on the event E.
Let F denote the set of queues whose length is $\xi_B(t-1)+1$ after incorporating arrivals for that slot. Let $F^{(i)}$ denote the updated set after i rounds of channel allocation. If $\xi_B(t) > \xi_B(t-1)$, then there exist at least $n(1-p-\epsilon)$ channels that were not used.

$$\begin{aligned} P(n(1-p-\epsilon) \text{ unused channels}) &= (1-q_2)^{\tilde{R}n^2(1-p-\epsilon)} \\ &= e^{-c_B n^2}, \end{aligned}$$

where $c_B = \tilde{R}(1-p-\epsilon)\log\frac{1}{1-q_2}$. Therefore,

$$P(\xi_B(t) > \xi_B(t-1)) \leq e^{-c_B n^2 + k(\epsilon)n} + e^{-nH(p|p+\epsilon)}.$$

■

We now prove there exists a constant k_0 such that the maximum relay queue-length decreases by 1 in k_0 consecutive time-slots with probability $\geq 1/2$.

Lemma 11. *We can find k_0 such that*

$$P(\xi_R(t+k_0) = \xi_R(t) - 1) \geq \frac{1}{2}.$$

Proof: The proof follows from Lemma 8 in [4] and Lemma 10 as stated above.

■

The following theorem uses the same proof technique as Theorem 5 in [5] to compute a bound on the rate function for the small buffer overflow event for the base-station queues using Lemma 10 and 11.

Theorem (5a)**.** *Under Assumption 4, for the SSG MaxWeight algorithm, for any $\epsilon \in (0, 1-p)$,*

$$\liminf_{n\to\infty} \frac{-1}{n} \log P\Big(\max_{1\leq i\leq n} Q_i(0) > b\Big) \geq c(b+1).$$

Where,

$$c = H(p|p+\epsilon) > 0.$$

Proof: Using Lemma 10 and Lemma 11 as stated above and by Theorem 5 in [5].

■

Relay Queues

In the following theorem we use the same proof technique as was used to compute the rate function of the SSG algorithm in [5] with the additional step that we use the fact that the base-station queues have less than b at the end of every time-slot with an exponentially large probability. Conditioned on this event, the maximum number of packets that arrive to any relay queue in a time-slot is $b+1$. This is an important step in this proof because potentially nS_{max} packets can arrive to a particular relay queue in a given time-slot and it is not possible to serve all of them in that time-slot and therefore the maximum queue-length in the relay queues can increase in a time-slot.

Theorem (5b)**.** *Under Assumption 4, for the SSG MaxWeight algorithm, for any $\epsilon \in (0, 1-p)$ and*

$$\delta \in \left(0, \frac{q_3(1-p-\epsilon)}{2-q_3}\right),$$

$$\liminf_{n\to\infty} \frac{-1}{n} \log P\left(\max_{1\le i\le n, 1\le j\le k} R_{ik}(0) > b\right) = (b+1)c_R,$$

where,

$$c_R \ge \min\left(H\big(p|p+\epsilon\big), \delta \log\frac{1}{1-q_3}, \frac{2\delta H\big(q_3|\frac{q_3}{2}\big)}{q_3}\right).$$

Proof: Consider the event E_5 that $\xi_B(t-1) = b$. This implies that all the base-station queues had less than b packets at the end of time-slot $t-1$. Then from Theorem 5a, we have that,

$$P(E_5^c) \le (b+1)s(n)e^{-nH(p|p+\epsilon)},$$

where $s(n)$ is a sub-exponential function of n. We condition the rest of the proof on the event E_5.
Conditioned on E_5, the maximum possible arrivals to any relay queue at the beginning of slot t is $b+1$. Therefore, using the same steps as in Theorem 5 in [5], we have that, for any $\epsilon \in (0, 1-p)$ and

$$\delta \in \left(0, \frac{q_3(1-p)}{2-q_3}\right),$$

$$\liminf_{n\to\infty} \frac{-1}{n} \log P\left(\max_{1\le i\le n, 1\le j\le k} R_{ik}(0) > b\right) = (b+1)c_R,$$

where,

$$c_R \ge \min\left(H\big(p|p+\epsilon\big), \delta \log\frac{1}{1-q_3}, \frac{2\delta H\big(q_3|\frac{q_3}{2}\big)}{q_3}\right).$$

■

C. ILQF BackPressure

For the ILQF BackPressure algorithm, we first focus on the relay queues and find the probability that in the steady state, they are all empty at the beginning of a slot. We observe that at the base-station, the iterative backpressure algorithm tries to serve queues with the highest backpressure values which are not always queues with maximum queue-lengths. However, conditioned on the fact that the relay queues are all empty, the two sets are the same. This allows us to bound the probability that the maximum base-station queue-length at the end of a time-slot is $> b$. Please refer to [14] for the complete proof.

D. ILQF MaxWeight

Similar to the analysis of the SSG MaxWeight algorithm, we first focus on the base-station queues and find the probability that in the steady state, they are have less than b packets at the beginning of a slot. Conditioned on the fact that the base-station queues have less than b packets at the end of time-slot $t-1$, not more than $b+1$ packet can arrive to any particular relay queue at the beginning of slot $t+1$. Using this, we find the probability that in the steady state, all relay queues are empty at the end of a time-slot. Please refer to [14] for the details.

Appendix C
k-hop Stability

We consider a $k-$hop full-duplex feed-forward network with 1 base-station, $k-1$ layers of relays and n users. The relays in the first layer of relays receive packets from the base-station and the relays in the k^{th} layer forward received packets to the users. A relay in the l^{th} layer (for $2 \le l \le k-1$) receives packets from the $(l-1)^{th}$ layer of relays and forwards them to the next layer. See Figure 5 for an example of such a network.

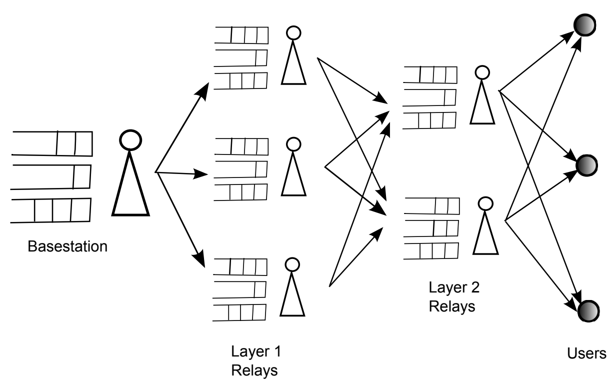

Fig. 5. An illustrative example of a 3-hop feed-forward relay network with 2 layers of relays and 3 users.

We use the following notation for this proof.

- $A_i(t)$ = the number of arrivals for user i at the base-station at the beginning of time-slot t.
- $Q_i(t)$ = The queue length of user i at the BS (measured at the end of the time-slot).
- $R_{(l),ri}(t)$ = The queue length of user i at relay r at layer l (measured at the end of the time-slot).
- $\mathbf{R}_{(l)}(t) = \{R_{(l)ri}(t) : \forall r; 1 \le i \le n\}$: The vector of queue lengths at the relays at layer l.

The $k-$hop version of the SSG MaxWeight algorithm is as follows:

In each time-slot, for each hop, sequentially allocate channels to queues in the following manner: first allocate channel S_1 to the maximum weight queue, i.e., the queue with largest queue-length channel-rate product. Then update the queue length based on the number of packets that are drained due to this allocation, and proceeds sequentially to the next channel (and so on).

For simplicity, we provide a proof of the stability of SSG MaxWeight under the following assumptions.

Assumption 5: (k-hop Stability)

- The base-station can forward packets to all relays in the first layer of relays. Each relay in layer l for every $l \in \{1, .., k-2\}$ can forward packets to all relays in the next layer (layer $l+1$). Each relay in layer $k-1$ can communicate with all the users.
- Bernoulli Arrivals and ON-OFF Channels
  - $A_i(t) = Bernoulli(p)$ i.i.d. across users and time-slots.
  - All channels are $Bernoulli(q)$ i.i.d. across channels, time-slots, relays and users.
- Linearly Scaling Relays: We assume that the l^{th} layer of relays has $\upsilon_l n$ relays for some constant $\upsilon_l > 0$.

Theorem 8. *Under Assumption 5, the $k-$hop system is stabilized by the SSG MaxWeight algorithm.*

Proof: The stability of the base-station queues follows from Lemma 5. In addition, by applying Theorem 4 for $\chi(n) = 1$ and $f(n) = e^{-nc_1}$ for some $c_1 > 0$, we have that

$$P(\max_i Q_i(t) > 0) \leq 4e^{-nc_1},$$

for all t.

Let F_1 be the event that $\max_i Q_i(t-1) = 0$. Therefore, we have that, $P(F_1^c) \leq 4e^{-nc_1}$. The rest of this proof is conditioned on F_1. Consider the queues at the relays of the first layer. In each round of channel allocation under the SSG MaxWeight algorithm, the probability that the channel cannot serve the currently longest queue (updated to reflect previous rounds of allocations) is $(1-q)^{\upsilon_2 n}$. Therefore with probability $> n(1-q)^{\upsilon_2 n}$, in a given time-slot, each channel serves the currently longest queue (updated to reflect previous rounds of allocations). Since the total arrivals to the relay queues at the first hop in a time-slot is less than $\leq n$, with probability $\geq 1 - 4e^{-nc_1} - n(1-q)^{\upsilon_2 n}$, the maximum queue-length at the first layer of relays does not increase in a time-slot. Therefore, we have that,

$$P(\max_{r,i} R_{(1)ri}(t+1) = \max_{r,i} R_{(1)ri}(t) + 1) \leq 4e^{-nc_1}.$$

Using this and Lemma 8 in [5], we can find k_0 such that,

$$P(\max_{r,i} R_{(1)ri}(t+1) = \max_{r,i} R_{(1)ri}(t) - 1) \geq \frac{1}{2}.$$

The stability of the relay queues at the first layer then follows using the Lyapunov function $Lyap(\mathbf{R}_{(1)}(t)) = \max_{r,i} R_{(1)ri}(t)$.

In addition, using Theorem 4, we have that,

$$P(\max_{r,i} R_{(1)ri}(t) > 0) \leq 16k_0 e^{-nc_1} + 4nk_0(1-q)^{\upsilon_2 n}.$$

For the queues at the l^{th} layer for $2 \leq l \leq k-2$, the proof of stability follows on the same lines as the proof of stability for relay queues at layer 1. For layer l, the proof follows by conditioning on the event that the queues at the base-station and relay layers 1 to $l-1$ are empty in the previous l time-slots.

The stability of the relay queues at layer l follows from Lemma 9, thus completing the proof of Theorem 8.

■