EURANDOM PREPRINT SERIES
2011-021

Instability of MaxWeight Scheduling Algorithms

P.M. van de Ven, S. Borst, S. Shneer
ISSN 1389-2355

Instability of MaxWeight Scheduling Algorithms*

Peter van de Ven[1,2], Sem Borst[1,3], Seva Shneer[1,2]

[1]Department of Mathematics & Computer Science
Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven, The Netherlands

[2]EURANDOM, P.O. Box 513, 5600 MB Eindhoven, The Netherlands

[3]Bell Laboratories, Alcatel-Lucent
P.O. Box 636, Murray Hill, NJ 07974, USA

April 25, 2011

Abstract

MaxWeight scheduling algorithms provide an effective mechanism for achieving queue stability and guaranteeing maximum throughput in a wide variety of scenarios. The maximum-stability guarantees however rely on the fundamental premise that the system consists of a fixed set of sessions with stationary ergodic traffic processes. In the present paper we examine a scenario where the population of active sessions varies over time, as sessions eventually end while new sessions occasionally start. We identify a simple necessary and sufficient condition for stability, and show that MaxWeight policies may fail to provide maximum stability. The intuitive explanation is that these policies tend to give preferential treatment to flows with large backlogs, so that the rate variations of flows with smaller backlogs are not fully exploited. In the usual framework with a fixed collection of flows, the latter phenomenon cannot persist since the flows with smaller backlogs will build larger queues and gradually start receiving more service. With a dynamic population of flows, however, MaxWeight policies may constantly get diverted to arriving flows, while neglecting the rate variations of a persistently growing number of flows in progress with relatively small remaining backlogs. We also perform extensive simulation experiments to corroborate the analytical findings.

1 Introduction

MaxWeight-type scheduling algorithms provide an effective mechanism for achieving maximum throughput and guaranteeing queue stability in a wide variety of scenarios. In a seminal paper, Tassiulas & Ephremides [23] presented a MaxWeight scheduling policy for throughput maximization in multi-hop wireless networks, where only certain subsets of the links may be simultaneously activated due to interference considerations, see also Kahale & Wright [7] for instance. In subsequent work, Tassiulas & Ephremides [24] described a MaxWeight policy for allocating a server among several parallel queues with time-varying connectivity.

Broadening the latter framework, MaxWeight-type policies were developed for power control and scheduling of shared wireless downlink channels with rate variations, see for instance Andrews *et al.* [2], Neely [13] and Neely *et al.* [15, 16]. Extending the scope further, Eryilmaz & Srikant [4], Neely *et al.* [14] and Stolyar [20, 21] devised algorithms for joint congestion control, routing and

*The work was financially supported by the EC under project number FP6-IST-034413.

scheduling based on MaxWeight principles. The powerful properties of MaxWeight-type policies have emerged as one the central paradigms in the broader realm of cross-layer control and resource allocation in wireless networks, see Georgiadis *et al.* [6] for a comprehensive overview.

MaxWeight-type algorithms have also been proposed for throughput maximization in input-queued switches, where only certain subsets of input-output pairs (e.g. matchings) may be simultaneously connected because of compatibility constraints, see for instance McKeown *et al.* [10]. The book of Meyn [11] contains extensive background material on MaxWeight policies. Crucial heavy-traffic results for MaxWeight algorithms were obtained by Stolyar [19].

The distinguishing characteristic of MaxWeight policies is that the subset of queues that are simultaneously served is selected so as to be of maximum 'weight', hence the term 'MaxWeight'. The weight of a queue is usually defined as the current backlog or the product of the backlog and the feasible instantaneous service rate for that queue, if selected. In multi-hop settings, the backlog differential is typically used, i.e., the difference in backlog with a downstream queue, giving rise to so-called back-pressure mechanisms. The combinations of queues which can be scheduled simultaneously are subject to certain constraints, based on for example interference conditions. In a more general sense, MaxWeight policies can be interpreted as selecting a service vector from a (possibly time-varying) feasible region that maximizes the inner product with the backlog vector.

Under mild assumptions, MaxWeight-type algorithms have been shown to provide maximum throughput, i.e., achieve queue stability whenever feasible to do so at all. A particularly appealing feature is that MaxWeight policies only need information on the current backlogs and instantaneous service rates, and do not rely on any explicit knowledge of the rate distributions or the traffic parameters. On the downside, finding the maximum weight subset is often a challenging problem and potentially NP-hard, which is exacerbated in a distributed setting, where message passing and exchange of backlog information create a substantial communication overhead in addition to the computational burden. This issue is especially pertinent as the maximum weight problem generally needs to be solved at a very high pace, commensurate with the fast time scale on which scheduling algorithms tend to operate. In order to address this issue, Tassiulas [22], Eryilmaz *et al.* [5] and Chaporkar & Sarkar [3] showed that randomized policies involve less stringent requirements and yet suffice for achieving maximum stability. In addition, several authors have considered algorithms that solve the maximum weight problem in some approximate sense, and quantified the resulting penalty in guaranteed throughput, see for instance Lin & Shroff [8], Sharma *et al.* [17, 18] and Wu & Srikant [25, 26].

As mentioned above, MaxWeight-type policies have been shown to achieve maximum stability under fairly mild assumptions. A fundamental premise however is that the system consists of a fixed set of queues with stationary ergodic traffic processes. In reality, the collection of active queues dynamically varies, as sessions eventually end, while new sessions occasionally start. In many situations the assumption of a fixed set of queues is still a reasonable modeling convention, since the scheduling actions and packet-level queue dynamics tend to occur on a very fast time scale, on which the population of active sessions evolves only slowly. In other cases, however, sessions may be relatively short-lived, and the above time scale separation argument does not apply. The impact of flow-level dynamics over longer time scales is particularly relevant in assessing stability properties, as the notion of stability only has strict meaning over infinite time horizons.

Motivated by the above observations, we examine in the present paper the stability properties of MaxWeight scheduling policies in a scenario with flow-level dynamics. For transparency, we focus on a point-to-point shared wireless downlink channel with rate variations, and do not consider multi-hop scenarios. We will show that MaxWeight scheduling policies may fail to provide maximum stability in the presence of flow-level dynamics. The intuitive explanation is that MaxWeight policies tend to give preferential treatment to flows with large backlogs, even when their service rates are not particularly favorable, and thus the rate variations of flows with smaller backlogs are not fully exploited. Note that the preferential treatment in fact also applies in the absence of any flow-level dynamics. In that case the phenomenon cannot persist however since the flows with smaller backlogs will build larger queues and gradually start receiving more service, creating a counteracting force. In contrast, in the presence of flow-level dynamics, MaxWeight policies may constantly get diverted to arriving flows, while neglecting the rate variations of a persistently growing number of flows in progress with relatively small remaining backlogs, so the opposing effect is never triggered.

It is worth observing that the instability of MaxWeight policies is fundamentally different from the instability of the Proportional Fair scheduling strategy demonstrated by Andrews [1]. The latter phenomenon is an illustration of the fact that utility-based scheduling strategies (which do not consider backlog information) may generally fail to achieve packet-level stability, even in the absence of any flow-level dynamics.

It is further worth drawing a distinction with the result in Lin *et al.* [9] showing the stability of joint scheduling and congestion control algorithms in the presence of flow-level dynamics without relying on the conventional simplifying time scale separation argument. The main difference with the present paper lies in the additional congestion control and the absence of rate variations in [9]. Inspection of the results in the present paper suggests that conventional forms of congestion control would not prevent the kind of instability phenomenon that we observe. In other words, the root cause for the instability appears not to be the lack of congestion control, but the fact that the rate variations are not maximally exploited in the presence of flow-level dynamics.

The remainder of the paper is organized as follows. In Section 2 we present a detailed model description. In Section 3 we derive a simple necessary and sufficient condition for stability in the presence of flow-level dynamics. Section 4 establishes that the MaxWeight policy may fail to provide maximum stability by treating specific model instances where the latter necessary and sufficient condition is satisfied, yet MaxWeight scheduling does not keep the system stable. In Section 5 extensive simulation results are provided to confirm the analytical findings and to demonstrate that the instability may also occur in more complex scenarios which do not lend themselves to an analytical treatment. In Section 6 we make some concluding remarks.

2 Model description

We consider a single wireless link shared by K classes of flows. The system operates in a time-slotted fashion, and in each time slot at most one of the flows can be scheduled for transmission.

Denote by $A_k(t)$ the number of class-k flows starting in time slot t. We assume that $A_k(1), A_k(2), \dots$ are i.i.d. copies of some random variable A_k with mean $\alpha_k < \infty$.

Each of the flows generates some finite random amount of traffic. We distinguish between two scenarios for the traffic influx of the various flows: (i) instantaneous traffic bursts; and (ii) gradual traffic streams.

In case (i) each flow generates an instantaneous traffic burst upon arrival to the system. Denote by B_{ki} the size of the burst of the i-th class-k flow (in bits). We assume that $B_{k1}, B_{k2}, \dots$ are i.i.d. copies of some integer random variable B_k with $\mathbb{E}\{B_k\} < \infty$.

In case (ii), each flow starts a random finite activity period upon arrival to the system, during which it produces a gradual stream of traffic. Denote by D_{ki} the duration of the activity period of the i-th class-k flow (in slots). We assume that $D_{k1}, D_{k2}, \dots$ are i.i.d. copies of some integer random variable D_k with $\mathbb{E}\{D_k\} < \infty$. Denote by $F_{ki}(t)$ the amount of traffic in bits generated by the i-th class-k flow in time slot t. For notational convenience, we define $F_{ki}(t)$ for all t, but its value is only relevant if the i-th class-k flow is active. We assume that $F_{ki}(1), F_{ki}(2), \dots$ are i.i.d. copies of some integer random variable F_k with $\mathbb{E}\{F_k\} < \infty$, and that the traffic processes are independent among the various flows. Denote by $B_{ki} := \sum_{t=S_{ki}}^{S_{ki}+D_{ki}-1} F_{ki}(t)$ the total amount of traffic generated by the i-th class-k flow, with S_{ki} denoting its arrival time. By the above assumptions, $B_{k1}, B_{k2}, \dots$ are i.i.d. copies of an integer random variable B_k with mean $\mathbb{E}\{B_k\} = \mathbb{E}\{D_k\}\mathbb{E}\{F_k\} < \infty$.

Note that scenario (i) may be interpreted as a special case of scenario (ii) with $D_k \equiv 1$ and $F_k \equiv B_k$. For economy of notation, however, it is useful to classify scenario (i) as a separate case. In both scenarios, traffic may only start to be served in the next slot after it arrives. Flows leave the system as soon as all their bits have been transmitted (and no further bits are due to arrive in the case of gradual traffic streams). During the period between its arrival and departure, a flow is said to be present.

The feasible transmission rates of the various flows vary over time as a result of fading. Denote by $R_{ki}(t)$ the feasible transmission rate (in bits) of the i-th class-k flow if selected for transmission in time slot t. For notational convenience, we define $R_{ki}(t)$ for all t, but its value is only relevant if the i-th class-k flow is actually present in the system. We assume that $R_{ki}(1), R_{ki}(2), \dots$ are i.i.d. copies of some integer, positive random variable R_k, and that the feasible transmission rates are independent among the various flows. Define $R_k^{\max} := \sup\{r : \mathbb{P}\{R_k = r\} > 0\}$ as the maximum possible value of the transmission rate of class-k flows (possibly $R_k^{\max} = \infty$).

Define $\rho_k := \alpha_k \tau_k$, and $\rho := \sum_{k=1}^{K} \rho_k$, with $\tau_k := \mathbb{E}\{\lceil B_k / R_k^{\max} \rceil\}$ when $R_k^{\max} < \infty$ and $\tau_k := 1$ when $R_k^{\max} = \infty$. Thus τ_k represents the expected number of slots required for the service of a class-k flow when served at rate $R_k^{\max}$.

3 Necessary and sufficient stability condition

In this section we first establish a simple necessary condition for stability to be achievable, and then proceed to show that this is in fact also (nearly) sufficient. The system is said to be stable if it empties infinitely often.

Proposition 3.1 *The condition $\rho \leq 1$ is necessary for stability to be achievable.*

Proof. The expected number of slots required for the service of an arbitrary class-k flow is bounded from below by τ_k. Thus the rate at which class-k work enters the system is bounded from below by $\rho_k = \alpha_k \tau_k$, and the total rate at which work arrives is bounded from below by $\rho = \sum_{k=1}^{K} \rho_k$. The latter quantity may not exceed one in order for stability to be achievable. $\Box$

We proceed to show that the above condition is also (nearly) sufficient for stability to be achievable. This may be intuitively explained as follows. With a dynamic population of flows, there will always be a flow that has the maximum possible feasible rate with high probability when there are sufficiently many flows present in the system. In other words, whenever a flow gets selected for transmission, it can be served at the maximum possible rate with high probability. Thus the expected number of slots required for the service of an arbitrary class-k flow can be brought arbitrarily close to τ_k, so that the system can be stabilized for values of ρ arbitrarily close to 1.

Evidently, the above explanation only provides heuristic arguments and does not account for several subtle yet critical issues. However, the intuitive insight offers useful guidance for the construction of a Lyapunov function which serves as the basis of a rigorous proof of the propositions presented below.

We will distinguish between the two traffic scenarios described in the previous section. As mentioned earlier, the scenario with instantaneous traffic bursts may be interpreted as a special case of that with gradual traffic streams. For transparency, however, we provide a separate treatment which introduces the key concepts while avoiding some of the additional complexity that arises in the general case. The proofs for both cases can be found in the appendix.

Proposition 3.2 *For any $\rho < 1$, there exists a scheduling strategy that achieves stability in case of* instantaneous *traffic.*

Proposition 3.3 *For any $\rho < 1$, there exists a scheduling strategy that achieves stability in case of* gradual *traffic.*

Remark 3.1 *It is worth emphasizing that the scheduling strategies considered in the proofs of Propositions 3.2 and 3.3 mainly serve to prove that $\rho < 1$ is sufficient for the existence of a stable strategy, and are therefore specifically designed for that purpose. The strategies may not be ideal for practical purposes as they may not provide particularly good performance, especially at lower loads. They also involve knowledge of various parameter values, which may be hard to obtain and is not used by the MaxWeight policy. (While the latter may be considered 'unfair', observe that in the standard case with a fixed set of flows no amount of additional information can help to* *achieve better stability performance than the MaxWeight policy provides.) The fact that for gradual traffic the scheduling strategy assumes prior knowledge of the duration of the activity period further adds to this. The design and analysis of suitable scheduling algorithms which guarantee maximum stability in the presence of flow dynamics while requiring minimum information and providing good performance across a wide range of loads, remains a challenging subject for further research.*

4 Instability of MaxWeight scheduling

In this section we establish that MaxWeight scheduling may fail to provide maximum stability. Specifically, we analyze two model instances where the sufficient condition stated in the previous section is satisfied, yet the MaxWeight strategy does not keep the system stable. For the sake of tractability, we focus on relatively simple models with instantaneous traffic and just a single class of flows. In the next section we present extensive simulation results to demonstrate that the instability may also occur in more complex scenarios with gradual traffic that do not lend themselves easily to an analytical treatment.

4.1 Scenario I

Flows start according to a Bernoulli process, i.e., in each time slot either a flow starts with probability α or no flow starts with probability $1-\alpha$, independent from slot to slot. The service requirement of each flow is a constant $B = 2D+1$ for some integer $D \geq 1$. The feasible transmission rate of a flow is either $D+1$ with probability p or $2D+1$ with probability $1-p$, $0 < p < 1$. The feasible transmission rates are independent across time and among different flows.

Proposition 3.2 states that $\rho = \alpha < 1$ is a sufficient condition for stability to be achievable. We now show that the MaxWeight scheduling strategy fails to achieve stability for $\rho = \alpha > 1/(1+p)$. The reason for the potential instability may be explained as follows. When a flow starts, the MaxWeight strategy will immediately serve it in the next slot, regardless of whether it has feasible rate $D+1$ or $2D+1$. To see that, observe that older flows present in the system will necessarily be of size D, and have no chance to be selected in competition with a new flow of size $2D+1$. In case the new flow has feasible rate $D+1$, it will require an additional slot at some later point for the service to be completed. In other words, the MaxWeight strategy 'wastes' a second slot on the service of flows whose initial feasible rate is $D+1$, whereas a single slot would suffice under a more cautious strategy. More specifically, since the expected number of slots required per flow is $1+p$, it follows that $\alpha > 1/(1+p)$ precludes stability.

Remark 4.1 *We can extend the example of instability to a slightly more general setting. Consider, as in the situation described above, a system with a single class of flows. Flows start according to a Bernoulli process, i.e., in each time slot either a flow starts with probability α or no flow starts with probability $1-\alpha$, independent from slot to slot. The service requirement of each flow is a constant B. In addition to $R^{\max}$, we also introduce $R^{\min} = \min\{i : \mathbb{P}\{R = i\} > 0\}$. Assume now that feasible service rates are such that*

$$(B - R^{\min}) \cdot R^{\max} < B \cdot R^{\min}.$$

It is easy to see that this condition implies that a flow entering the system will immediately get scheduled. Hence, the average number of slots required for the service of an arbitrary flow is bounded from below by

$$1 + \sum_{i=1}^{R^{\max}} \left\lceil \frac{B-i}{R^{\max}} \right\rceil \mathbb{P}\{R = i\}. \tag{1}$$

Thus, stability is precluded if

$$\alpha \left(1 + \sum_{i=1}^{R^{\max}} \left\lceil \frac{B-i}{R^{\max}} \right\rceil \mathbb{P}\{R = i\} \right) > 1.$$

Note that the quantity in (1) is strictly smaller than $\lceil B/R^{\max} \rceil$, provided that $R^{\min} < R^{\max}$.

4.2 Scenario II

We discuss a second scenario where the MaxWeight strategy fails to achieve maximum stability. As before, flows start according to a Bernoulli process, i.e., in each time slot either a flow starts with probability α or no flow starts with probability $1 - \alpha$, independent from slot to slot. The service requirement of each flow is a constant B. For convenience, we assume $B = 8D$ for some integer $D \geq 1$. The feasible transmission rate of a flow is either 1 with probability p or 2 with probability $1 - p$, $0 < p < 1$. The feasible transmission rates are independent across time and among different flows. In this case, Proposition 3.2 states that stability can be achieved as long as $\rho = 4\alpha D < 1$.

Let $N_i(t)$ denote the number of flows of size i at time t. It may be shown that for $\rho \leq 1$, the process $(N_{3D+1}(t), N_{3D+2}(t), \ldots, N_B(t))$ of flows of size $3D + 1$ or greater is 'stable'. This makes sense since large flows receive priority, and the onset of instability manifests itself in the growth of the number of small flows. It then follows that the system spends a non-negligible fraction of time in states where all flows of size $3D + 1$ or greater have rate 1 and there is at least one flow of size greater than $6D + 1$. In these states, the MaxWeight strategy will serve a flow at rate 1. Similar to the previous scenario, this means that the fraction of time that transmission rate 1 is used, does not approach 0 as $\rho \to 1$, and instability follows.

5 Numerical experiments

In this section we present simulation results that confirm the instability of MaxWeight scheduling, as well as clarify the nature of the instability. All simulations consist of a single run of 10^5 time slots. In each slot, a new flow starts with probability α.

The first scenario we consider is Scenario II from Section 4, with $D = 2$. Figure 1 shows the number of bits in the system, plotted for various values of α. Although the condition $\alpha < 1$ ensures the existence of a stable scheduling strategy in this scenario, it is easily seen that this is not sufficient for the MaxWeight policy to achieve stability.

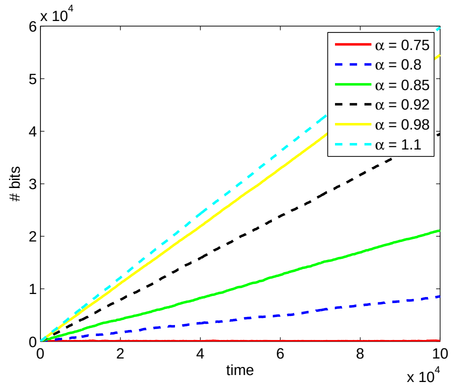
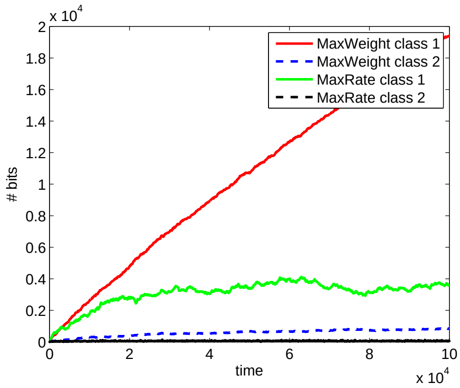
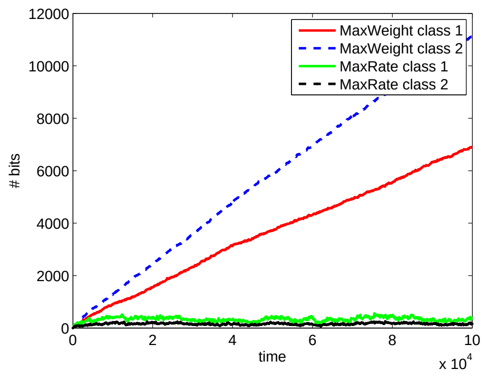


Figure 1: The number of bits in the system plotted against time under MaxWeight scheduling, for various values of α.

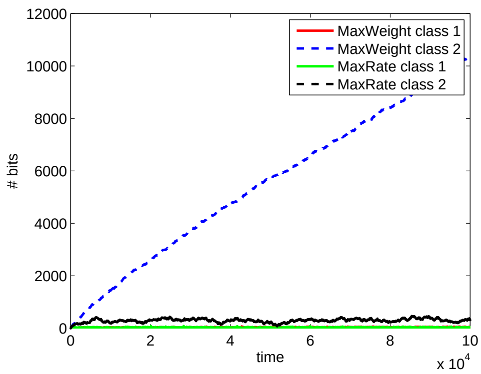
From this point on, we consider gradual traffic. During the activity period of a flow, each slot a single bit enters. The length of this period is geometrically distributed with parameter p. In Figure 2, three two-class scenarios are presented. Flows belong to either of the classes with equal probability, and the transmission rates are geometrically distributed with parameter q. Hence, $R^{\max} = \infty$, and the necessary stability condition found in Proposition 3.1 simplifies to $\alpha < 1$. Besides the sample path for MaxWeight scheduling, we also plot the behavior of MaxRate scheduling, a somewhat simpler version of the algorithm used in Propositions 3.2 and 3.3, in which the flow with the highest rate is scheduled. In each of these figures, MaxRate scheduling provides stability, whereas MaxWeight scheduling fails to do so. Note that although the MaxWeight scheduling policy is unstable in the cases presented, it is still possible for particular classes of flows to be stable. This is in contrast to MaxWeight scheduling in the static scenario.



(a) $p_1 = p_2 = 1/4$, $q_1 = 1/2$, $q_2 = 1/4$, $\alpha = 0.93$.



(b) $p_1 = 1/4$, $p_2 = 1/2$, $q_1 = q_2 = 1/2$, $\alpha = 0.95$.



(c) $p_1 = 1/4$, $p_2 = 1/2$, $q_1 = 1/4$, $q_2 = 1/2$, $\alpha = 0.95$.

Figure 2: The number of bits in the system of both classes plotted against time for various parameters.

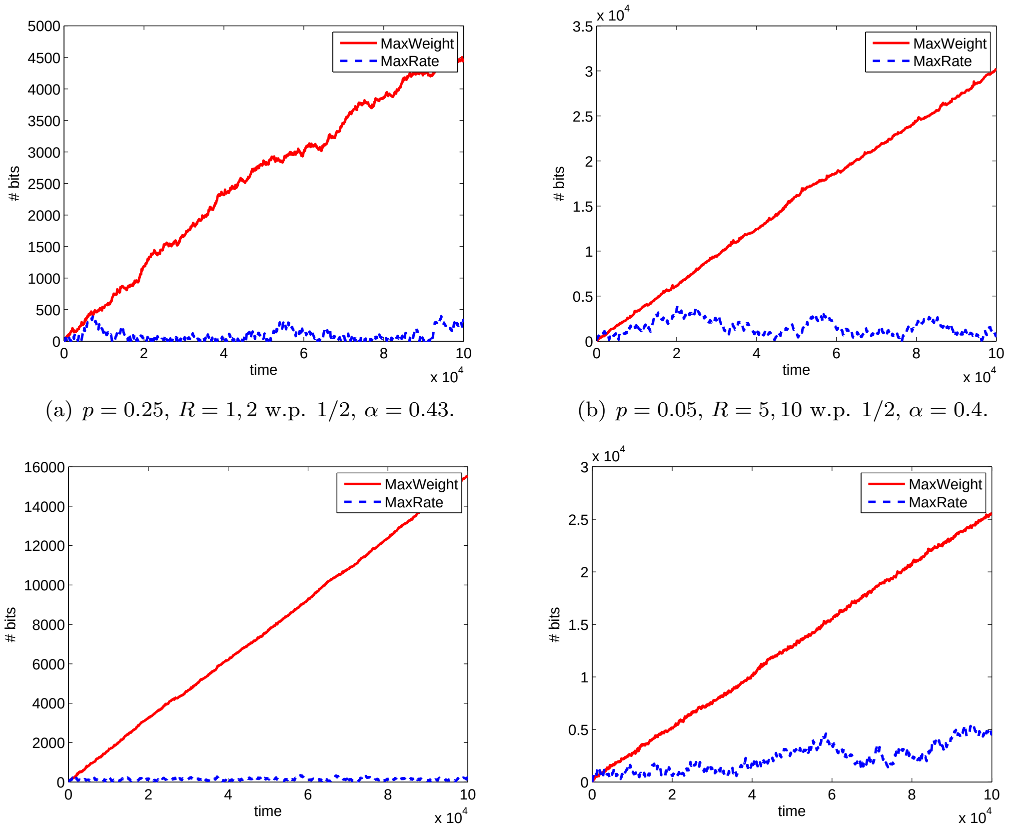
Figure 3 displays the number of bits over time in a single-class scenario when the transmission rates can assume only two possible values.

Figure 4 contains a similar scenario, but with the transmission rates geometrically distributed with parameter q, so $R^{\max} = \infty$. This figure again demonstrates that MaxWeight fails to provide maximum stability.

6 Conclusion

We studied the performance of MaxWeight scheduling in a setting where flow dynamics are taken into consideration. We determined an explicit necessary condition for stability, and devised a simple policy to show that this condition in fact is also (nearly) sufficient for stability. Two illustrative examples were provided of scenarios where MaxWeight scheduling fails to attain stability under this condition.

The analytical results are supported and complemented by simulation experiments for more involved scenarios. The simulations compare the MaxWeight scheduling strategy to the MaxRate



(a) $p = 0.25,\ R = 1, 2$ w.p. $1/2,\ \alpha = 0.43$.

(b) $p = 0.05,\ R = 5, 10$ w.p. $1/2,\ \alpha = 0.4$.

(c) $p = 0.25,\ R = 1, 4$ w.p. $9/10,\ 1/10,\ \alpha = 0.65$.

(d) $p = 0.05,\ R = 5, 20$ w.p. $9/10,\ 1/10,\ \alpha = 0.64$.

Figure 3: The number of bits in the system plotted against time for various parameters.

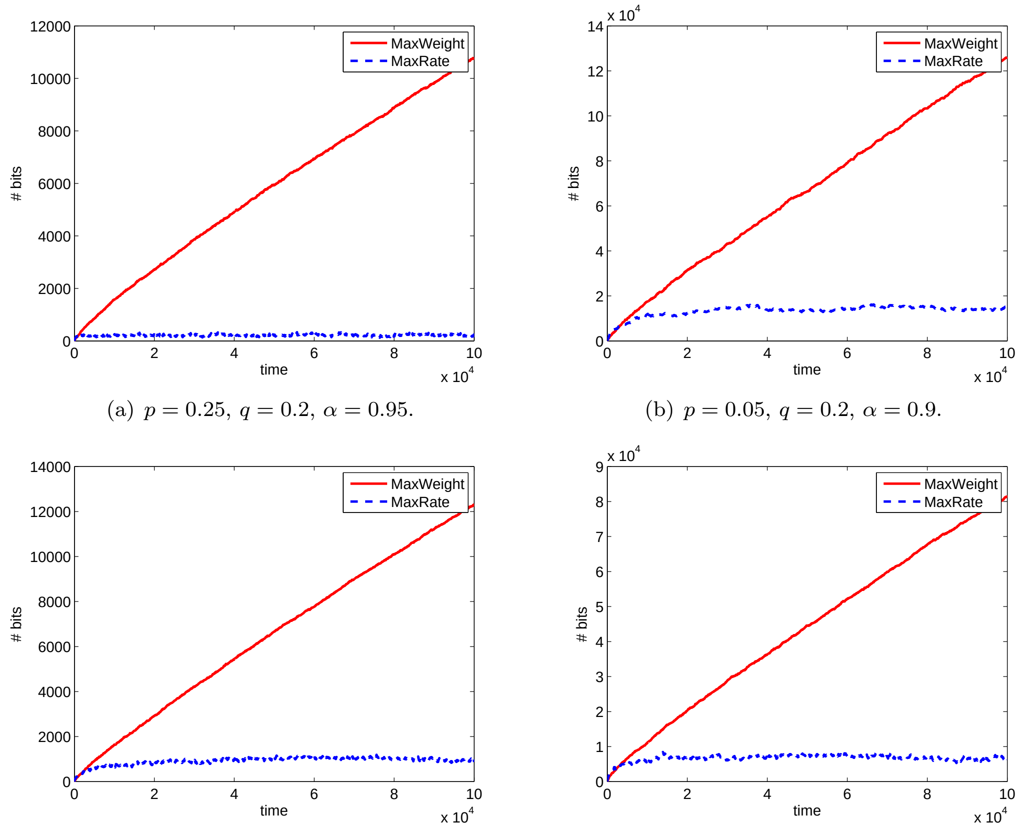
policy, and confirm the instability of the MaxWeight strategy.

It is worth recalling that although the policies that we constructed in the proofs of Propositions 3.2 and 3.3 achieve maximum stability, they may not be ideal for practical purposes and may not provide particularly good delay performance, especially at lower loads. Moreover, the policies require knowledge of various parameter values which may be difficult to obtain in practical situations and is not used by the MaxWeight policy. (While the latter may be considered 'unfair', note however that in the standard case with a fixed set of flows no amount of additional information can help to achieve better stability performance than the MaxWeight policy provides.) The design and analysis of suitable scheduling algorithms which guarantee maximum stability in the presence of flow dynamics remains a challenging subject for further research.

Proof of Proposition 3.2. We first introduce several constants that will be used. Let $\epsilon := \frac{1}{2}(1-\rho)/(K+1) > 0$. Denote $\theta_k := [\mathbb{P}\{R_k > 0\}]^{-1} < \infty$, let $L_k := \min\{l : \sum_{i=l+1}^{\infty} i\mathbb{P}\{B_k = i\} \leq \epsilon/(\alpha_k\theta_k)\}$, and observe that $L_k < \infty$ since $\mathbb{E}\{B_k\} < \infty$. Define $Z_k = \min\{R_k^{\max}, \alpha_k\mathbb{E}\{B_k\}/\epsilon\}$. Note that $\alpha_k\mathbb{E}\{\lceil B_k/Z_k\rceil\} \leq \rho_k + \epsilon$.

We consider a scheduling strategy with the following property: it serves a class-k flow that either (i) has a feasible transmission rate Z_k or higher or (ii) has a residual size L_k or larger and a positive feasible transmission rate, whenever possible. Ties are broken arbitrarily. Let us say that in time slot t a flow of class $k(t)$ with a residual size of $l(t)$ bits is served at rate $r(t)$, with the convention that $k(t) = l(t) = r(t) = 0$ in case no flow gets scheduled in time slot t at all.

In order to describe the evolution of the system over time, we introduce the process $N(t) = (N_1(t), \dots, N_K(t))$, with $N_k(t) = (N_1^k(t), N_2^k(t), \dots)$ and $N_l^k(t)$ representing the number of class-k flows in the system with a residual size of l bits at the beginning of slot t.



(a) $p = 0.25$, $q = 0.2$, $\alpha = 0.95$.

(b) $p = 0.05$, $q = 0.2$, $\alpha = 0.9$.

(c) $p = 0.25$, $q = 0.05$, $\alpha = 0.99$.

(d) $p = 0.05$, $q = 0.05$, $\alpha = 0.95$.

Figure 4: The number of bits in the system plotted against time for various parameters.

Observe that

$$\begin{aligned} N_l^k(t+1) = & N_l^k(t) + A_l^k(t) - \mathrm{I}_{\{k(t)=k, l(t)=l\}} \\ & + \mathrm{I}_{\{k(t)=k, l(t)=l+r(t)\}}, \end{aligned}$$

with $A_l^k(t)$ denoting the number of class-k flows arriving at time t with a size of exactly l bits. It is easily verified that the process $N(t)$ is a Markov chain.

Define the Lyapunov function:

$$V(n) := \sum_{k=1}^{K} \left(\sum_{i=1}^{L_k} n_i^k \left\lceil \frac{i}{Z_k} \right\rceil + \theta_k \sum_{i=L_k+1}^{\infty} i n_i^k \right),$$

with $n = (n_1, \ldots, n_K)$ and $n_k = (n_1^k, n_2^k, \ldots)$.

The above function provides a measure for the total amount of work in the system in terms of the total number of slots required for the service of all currently present flows, assuming that class-k flows of residual size no larger than L_k are always served at rate Z_k, while class-k flows of residual size of at most L_k are served at rate $\theta_k^{-1} = \mathbb{P}\{R_k > 0\}$.

We can write the drift as

$$V(N(t+1)) - V(N(t)) = \sum_{k=1}^{K} I_k(t) - D(t),$$

with

$$I_k(t) := \sum_{i=1}^{L_k} A_i^k(t) \left\lceil \frac{i}{Z_k} \right\rceil + \theta_k \sum_{i=L_k+1}^{\infty} i A_i^k(t),$$

reflecting the increase in the workload due to the arrival of class-k flows, and

$$\begin{aligned}D(t) :=& \left\lceil \frac{l(t)}{Z_{k(t)}} \right\rceil \mathrm{I}_{\{1 \le l(t) \le L_{k(t)}\}} + \theta_{k(t)} l(t) \mathrm{I}_{\{l(t) > L_{k(t)}\}} \\ &- \left\lceil \frac{l(t) - r(t)}{Z_{k(t)}} \right\rceil \mathrm{I}_{\{1 \le l(t) - r(t) \le L_{k(t)}\}} \\ &- \theta_{k(t)} (l(t) - r(t)) \mathrm{I}_{\{l(t) - r(t) > L_{k(t)}\}}\end{aligned}$$

representing the decrease in the workload due to the service of flows.

The conditional drift may then be written as:

$$\begin{aligned}&\mathbb{E}\{V(N(t+1)) - V(N(t)) \mid N(t) = n\} \\ =& \sum_{k=1}^{K} \mathbb{E}\{I_k(t)\} - \mathbb{E}\{D(t) \mid N(t) = n\}.\end{aligned} \tag{2}$$

We first derive an upper bound for $\mathbb{E}\{I_k(t)\}$.

$$\begin{aligned}&\mathbb{E}\{I_k(t)\} \\ =& \sum_{i=1}^{L_k} \left\lceil \frac{i}{Z_k} \right\rceil \mathbb{E}\{A_i^k(t)\} + \theta_k \sum_{i=L_k+1}^{\infty} i \mathbb{E}\{A_i^k(t)\} \\ =& \alpha_k \left(\sum_{i=1}^{L_k} \left\lceil \frac{i}{Z_k} \right\rceil \mathbb{P}\{B_k = i\} + \theta_k \sum_{i=L_k+1}^{\infty} i \mathbb{P}\{B_k = i\} \right) \\ =& \alpha_k \left(\mathbb{E}\{\left\lceil \frac{B_k}{Z_k} \right\rceil\} + \theta_k \sum_{i=L_k+1}^{\infty} (i - \left\lceil \frac{i}{Z_k} \right\rceil / \theta_k) \mathbb{P}\{B_k = i\} \right) \\ \le& \rho_k + 2\epsilon.\end{aligned} \tag{3}$$

We now turn to a lower bound for $\mathbb{E}\{D(t) \mid N(t) = n\}$. Note that

$$\begin{aligned}D(t) \ge& \mathrm{I}_{\{1 \le l(t) \le L_{k(t)}, r(t) \ge Z_{k(t)}\}} + \theta_{k(t)} \mathrm{I}_{\{l(t) > L_{k(t)}, r(t) > 0\}} \\ =& \mathrm{I}_{\{1 \le l(t) \le L_{k(t)}, r(t) \ge Z_{k(t)}\}} \\ &+ [\mathbb{P}\{R_{k(t)} > 0\}]^{-1} \mathrm{I}_{\{l(t) > L_{k(t)}, r(t) > 0\}}.\end{aligned}$$

Now, let $E^{small}(t)$ be the event that there is at least one class-k flow in time slot t of residual size no larger than L_k with feasible transmission rate Z_k or higher. Let $E^{large}(t)$ be the event that there is at least one class-k flow in time slot t of residual size $L_k + 1$ or larger with a non-zero feasible transmission rate.

Note that

$$\begin{aligned}&\mathrm{I}_{\{l(t) > L_{k(t)}, r(t) > 0\}} \\ =& \mathrm{I}_{\{l(t) \ge L_{k(t)} + 1, r(t) > 0, E^{small}(t)\}} + \mathrm{I}_{\{l(t) > L_{k(t)}, r(t) > 0, \bar{E}^{small}(t)\}} \\ =& \mathrm{I}_{\{l(t) > L_{k(t)}, r(t) > 0, E^{small}(t)\}} + \mathrm{I}_{\{E^{large}(t), \bar{E}^{small}(t)\}}.\end{aligned}$$

Further observe

$$\begin{aligned}&\mathrm{I}_{\{l(t) \le L_{k(t)}, r(t) \ge Z_{k(t)}\}} + \mathrm{I}_{\{l(t) > L_{k(t)}, E^{small}(t)\}} \\ =& \mathrm{I}_{\{E^{small}(t)\}} = 1 - \mathrm{I}_{\{\bar{E}^{small}(t)\}}.\end{aligned}$$

We deduce that

$$D(t) \geq 1 - \mathrm{I}_{\{\bar{E}^{small}(t)\}} + [\mathbb{P}\{R_{k(t)} > 0\}]^{-1}\mathrm{I}_{\{E^{large}(t),\bar{E}^{small}(t)\}}.$$

Thus,

$$\begin{aligned}
&\mathbb{E}\{D(t)|N(t) = n\} \\
=&1 - \mathbb{P}\{\bar{E}^{small}(t)|N(t) = n\} + [\mathbb{P}\{R_{k(t)} > 0\}]^{-1} \\
&\cdot \mathbb{P}\{E^{large}(t), \bar{E}^{small}(t)|N(t) = n\} \\
=&1 - \mathbb{P}\{\bar{E}^{small}(t)|N(t) = n\} \\
&\cdot (1 - [\mathbb{P}\{R_{k(t)} > 0\}]^{-1}\mathbb{P}\{E^{large}(t)|N(t) = n\}).
\end{aligned}$$

Let $s_k := \sum_{i=L_k}^{\infty} n_i^k$. If $\sum_{k=1}^{K} s_k > 0$, then $\mathbb{P}\{E^{large}(t)|N(t) = n\} \geq \min_{k=1,\dots,K} \mathbb{P}\{R_k > 0\}$, so that $[\mathbb{P}\{R_{k(t)} > 0\}]^{-1}\mathbb{P}\{E^{large}(t)|N(t) = n\} \geq 1$.

If $\sum_{k=1}^{K} n_k \geq N_{\epsilon,\eta}$, then $\mathbb{P}\{\bar{E}^{small}(t)|N(t) = n\} \leq (1-\eta)^{N_{\epsilon,\eta}} \leq \epsilon$.

Define

$$C = \{n| \sum_{k=1}^{K} n_k < N_{\epsilon,\eta} \text{ and } \sum_{k=1}^{K} s_k = 0\}.$$

Then we obtain that

$$\mathbb{E}\{D(t) \mid N(t) = n\} \geq 1 - \epsilon \tag{4}$$

for any $x \notin C$.

Combining Equations (2), (3), and (4),

$$\mathbb{E}\{V(N(t+1)) - V(N(t)) \mid N(t) = n\} \leq -\epsilon,$$

for any $n \notin C$. In addition, it is easily verified that $\mathbb{E}\{V(N(t+1))|N(t) = m\} < \infty$ for any $m \in C$.

Inspection of the Foster-Lyapunov drift criteria then shows that the set C is visited infinitely often [12]. Since C is finite and the all-empty state is reachable from any state in C, if we additionally assume the scheduling strategy to be non-idling, it follows that the system must empty infinitely often. □

Remark .1 *If B_k has finite support, i.e., $B_k^{\max} := \sup\{b : \mathbb{P}\{B_k = b\} > 0\} < \infty$, then the above proof may be considerably simplified by taking $L_k = B_k^{\max}$ and dropping all the terms involving N_l^k, $l \geq B_k^{\max} + 1$.*

Proof of Proposition 3.3. We introduce several constants that will be used. Let $\epsilon := \frac{1}{2}(1-\rho)/(K+2) > 0$ and define θ_k, L_k and Z_k as in the proof of Proposition 3.2. In addition to that, define $\sigma_k := \alpha_k \mathbb{E}\{D_k\}$, $\sigma := \sum_{k=1}^{K} \sigma_k$, $\delta = \epsilon/\sigma$, and $\Omega = (1-\epsilon)/\delta$. Finally, let $M_k := \min\{m : \sum_{j=m+1}^{\infty} j\mathbb{P}\{D_k = j\} \leq \sigma_k/(\alpha_k\Omega)\}$, and observe that $M_k < \infty$ since $\mathbb{E}\{D_k\} < \infty$.

We consider a scheduling strategy with the following property: it serves an inactive class-k flow that either (i) has a feasible transmission rate Z_k or higher; or (ii) has a residual size greater than L_k and a positive feasible transmission rate, whenever possible. Let us say that in time slot t a flow of class $k(t)$ with a residual size of $l(t)$ bits is served at rate $r(t)$, with the convention that $k(t) = l(t) = r(t) = 0$ in case no flow gets scheduled in time slot t at all.

In order to describe the evolution of the system over time, we introduce $N(t) = (N_1(t), \dots, N_K(t))$, with $N_k(t) = (N_1^k(t), Q_1^k(t), N_2^k(t), Q_2^k(t), \dots)$, $Q_l^k(t) = (Q_{l1}^k(t), Q_{l2}^k(t), \dots)$, $N_l^k(t)$ representing the number of inactive class-k flows in the system at the beginning of slot t with a residual size of l bits, and $Q_{lm}^k(t)$ the number of class-k flows in the system at time t with a residual activity period of length m and a total size of l bits.

Observe that

$$N_l^k(t+1) = N_l^k(t) + Q_{l1}^k(t) - \mathrm{I}_{\{k(t)=k, l(t)=l\}} \\ + \mathrm{I}_{\{k(t)=k, l(t)=l+r(t)\}},$$

and

$$Q_{lm}^k(t+1) = Q_{lm+1}^k(t) + A_{lm}^k(t),$$

with $A_{lm}^k(t)$ denoting the number of class-k flows arriving at time t with an activity period of length m and a size of exactly l bits. It is easily verified that the process $N(t)$ is a Markov chain.

Define the Lyapunov function:

$$V(n) := \sum_{k=1}^{K} \left(\delta \sum_{j=1}^{M_k} j q_{*j}^k + \delta\Omega \sum_{j=M_k+1}^{\infty} j q_{*j}^k \right. \\ \left. + \sum_{i=1}^{L_k} (q_{i*}^k + n_i^k) \left\lceil \frac{i}{Z_k} \right\rceil + \theta_k \sum_{i=L_k+1}^{\infty} i (q_{i*}^k + n_i^k) \right),$$

with $n = (n_1, \ldots, n_K)$, $n_k = (n_1^k, q_1^k, n_2^k, q_2^k, \ldots)$, $q_l^k = (q_{l1}^k, q_{l2}^k, \ldots)$, $q_{l*}^k = \sum_{j=1}^{\infty} q_{lj}^k$, $q_{*m}^k = \sum_{i=1}^{\infty} q_{im}^k$. The above function provides a measure for the total workload and weighted aggregate residual lifetime of all the flows present in the system.

Note that

$$V(N(t+1)) - V(N(t)) = \sum_{k=1}^{K} I_k(t) + \delta \sum_{k=1}^{K} J_k(t) \\ - \delta \sum_{k=1}^{K} E_k(t) - D(t), \tag{5}$$

with

$$I_k(t) := \left(\sum_{i=1}^{L_k} A_{i*}^k(t) \left\lceil \frac{i}{Z_k} \right\rceil + \theta_k \sum_{i=L_k+1}^{\infty} i A_{i*}^k(t) \right),$$

reflecting the increase in the workload due to the arrival of class-k flows,

$$J_k(t) := \sum_{j=1}^{M_k} j A_{*j}^k(t) + \Omega \sum_{j=M_k+1}^{\infty} j A_{*j}^k(t),$$

with $A_{i*}^k(t) := \sum_{j=1}^{\infty} A_{ij}^k(t)$, $A_{*j}^k(t) := \sum_{i=1}^{\infty} A_{ij}^k(t)$, representing the increase in the aggregate residual lifetime due to the arrival of class-k flows,

$$D(t) := \lceil \frac{l(t)}{Z_{k(t)}} \rceil \mathrm{I}_{\{1 \leq l(t) \leq L_{k(t)}\}} + \theta_{k(t)} l(t) \mathrm{I}_{\{l(t) > L_{k(t)}\}} \\ - \lceil \frac{l(t) - r(t)}{Z_{k(t)}} \rceil \mathrm{I}_{\{1 \leq l(t) - r(t) \leq L_{k(t)}\}} \\ - \theta_{k(t)} (l(t) - r(t)) \mathrm{I}_{\{l(t) - r(t) > L_{k(t)}\}}$$

capturing the decrease in the workload due to the service of inactive flows, and

$$E_k(t) := \sum_{j=1}^{M_k} Q_{*j}^k(t) + \Omega \sum_{j=M_k+1}^{\infty} Q_{*j}^k(t)$$

corresponding to the decrease in the aggregate residual lifetime due to the aging of active class-k flows.

The conditional drift may then be written as:

$$\begin{aligned}
&\mathbb{E}\{V(N(t+1)) - V(N(t)) \mid N(t) = n\} \\
=&\sum_{k=1}^{K} \mathbb{E}\{I_k(t)\} + \delta \sum_{k=1}^{K} \mathbb{E}\{J_k(t)\} \\
&- \delta \sum_{k=1}^{K} \mathbb{E}\{E_k(t) \mid N(t) = n\} - \mathbb{E}\{D(t) \mid N(t) = n\}.
\end{aligned}$$

As the arrival process of new flows is the same in both settings, we conclude from Equation (3) that

$$\mathbb{E}\{I_k(t)\} \leq \rho_k + 2\epsilon. \tag{6}$$

Next we establish an upper bound for $\mathbb{E}\{J_k(t)\}$.

$$\begin{aligned}
&\mathbb{E}\{J_k(t)\} \\
=&\sum_{j=1}^{M_k} j\mathbb{E}\{A_{*j}^k(t)\} + \Omega \sum_{j=M_k+1}^{\infty} j\mathbb{E}\{A_{*j}^k(t)\} \\
=&\alpha_k \left(\sum_{j=1}^{M_k} j\mathbb{P}\{D_k = j\} + \Omega \sum_{j=M_k+1}^{\infty} j\mathbb{P}\{D_k = j\} \right) \\
=&\alpha_k \left(\mathbb{E}\{D_k\} + \Omega \sum_{j=M_k+1}^{\infty} j(1 - 1/\Omega)\mathbb{P}\{D_k = j\} \right) \\
\leq& 2\sigma_k.
\end{aligned} \tag{7}$$

We proceed with a lower bound for $\mathbb{E}\{E_k(t)|N(t) = n\}$.

$$\mathbb{E}\{E_k(t)|N(t) = n\} = \sum_{j=1}^{M_k} q_{*j}^k + \Omega \sum_{j=M_k+1}^{\infty} q_{*j}^k. \tag{8}$$

Thus $\mathbb{E}\{E_k(t)|N(t) = n\} \geq \Omega$ whenever $q_k := \sum_{j=1}^{M_k} q_{*j}^k \geq \Omega$ or $s'_k := \sum_{j=M_k+1}^{\infty} q_{*j}^k \geq 1$.

We turn to a lower bound for $\mathbb{E}\{D(t) \mid N(t) = n\}$. Define

$$C = \{n| \sum_{k=1}^{K} n_k < N_{\epsilon,\eta} \text{ and } \sum_{k=1}^{K} s_k = 0\}.$$

Using similar arguments as in the previous proof, we then obtain

$$\mathbb{E}\{D(t) \mid N(t) = n\} \geq 1 - \epsilon \tag{9}$$

for any $n \notin C$.

Define the set

$$\begin{aligned}
\hat{C} =&\{n| \sum_{k=1}^{K} n_k \leq N_{\epsilon,\eta} \text{ and } \sum_{k=1}^{K} s_k = 0 \\
&\text{and } \sum_{k=1}^{K} q_k \leq \Omega \text{ and } \sum_{k=1}^{K} s'_k = 0\}.
\end{aligned}$$

Suppose $n \notin \hat{C}$. Then either $\sum_{k=1}^{K} q_k > \Omega$ or $\sum_{k=1}^{K} s'_k \geq 1$ or $n \notin C$.
If $n \notin C$, then the conditional drift is bounded from above by

$$\rho + 2K\epsilon + 2\delta\sigma - 1 + \epsilon = \rho + (2K+3)\epsilon - 1 = -\epsilon.$$

If $\sum_{k=1}^{K} q_k > \Omega$ or $\sum_{k=1}^{K} s'_k \geq 1$, then the conditional drift is bounded from above by

$$\rho + 2K\epsilon + 2\delta\sigma - \delta\Omega = \rho + (2K+3)\epsilon - 1 = -\epsilon.$$

Combining Equations (5)-(9),

$$\mathbb{E}\{V(N(t+1)) - V(N(t)) \mid N(t) = n\} \leq -\epsilon,$$

for any $n \notin \hat{C}$. In addition, it is easily verified that $\mathbb{E}\{V(N(t+1))|N(t) = m\} < \infty$ for any $m \in \hat{C}$.
Inspection of the Foster-Lyapunov drift criteria then shows that the set $\hat{C}$ is visited infinitely often [12]. Since $\hat{C}$ is finite, and the all-empty state is reachable from any state in $\hat{C}$, if we additionally assume the scheduling strategy to be non-idling, it follows that the system must empty infinitely often. □

References

[1] D.M. Andrews (2004). Instability of the Proportional Fair scheduling algorithm for HDR. *IEEE Trans. Wireless Commun.* **3**, 1422–1426.

[2] D.M. Andrews, K. Kumaran, K. Ramanan, A.L. Stolyar, R. Vijayakumar, P.A. Whiting (2004). Scheduling in a queueing system with asynchronously varying service rates. *Prob. Eng. Inf. Sc.* **18**, 191–217.

[3] P. Chaporkar, S. Sarkar (2006). Stable scheduling policies for maximizing throughput in generalized constrained queueing. In: *Proc. Infocom 2006*.

[4] A. Eryilmaz, R. Srikant (2005). Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control. In: *Proc. Infocom 2005*, 1794–1803.

[5] A. Eryilmaz, R. Srikant, J. Perkins (2005). Stable scheduling policies for fading wireless channels. *IEEE/ACM Trans. Netw.* **13**, 411–424.

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

[7] N. Kahale, P.E. Wright (1997). Dynamic global packet routing in wireless networks. In: *Proc. Infocom '97*, 1416–1423.

[8] X. Lin, N.B. Shroff (2005). The impact of imperfect scheduling on cross-layer rate control in wireless networks. In: *Proc. Infocom 2005*.

[9] X. Lin, N.B. Shroff, R. Srikant (2008). On the connection-level stability of congestion-controlled communication networks. *IEEE Trans. Inf. Theory* **54**, 2317–2338.

[10] N. McKeown, V. Anantharam, J.C. Walrand (1996). Achieving 100% throughput in an input-queued switch. In: *Proc. Infocom '96*, 296–302.

[11] S.P. Meyn (2007). *Control Techniques for Complex Networks*, Cambridge University Press.

[12] S.P. Meyn, R.L. Tweedie (1993). *Markov Chains and Stochastic Stability*, Springer Verlag.

[13] M.J. Neely (2005). Energy optimal control for time-varying wireless networks. In: *Proc. Infocom 2005*, 572–583.

[14] M.J. Neely, E. Modiano, C.-P. Li (2005). Fairness and optimal stochastic control for heterogeneous networks. In: *Proc. Infocom 2005*.

[15] M.J. Neely, E. Modiano, C.E. Rohrs (2002). Power and server allocation in a multi-beam satellite with time-varying channels. In: *Proc. Infocom 2002*, 1451–1460.

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

[17] G. Sharma, R.R. Mazumdar, N.B. Shroff (2006). On the complexity of scheduling in wireless networks. In: *Proc. MobiCom '06*, 227–238.

[18] G. Sharma, N.B. Shroff, R.R. Mazumdar (2007). Joint congestion control and distributed scheduling for throughput guarantees in wireless networks. In: *Proc. Infocom 2007*.

[19] A.L. Stolyar (2004). MaxWeight scheduling in a generalized switch: state space collapse and workload minimization in heavy traffic. *Ann. Appl. Prob.* **14**, 1–53.

[20] A.L. Stolyar (2005). Maximizing queueing network utility subject to stability: greedy primal dual algorithm. *Queueing Systems* **50**, 401–457.

[21] A.L. Stolyar (2006). Greedy primal-dual algorithm for dynamic resource allocation in complex networks. *Queueing Systems* **54**, 203–220.

[22] L. Tassiulas (1998). Linear complexity algorithms for maximum throughput in radio networks and input queued switches. In: *Proc. Infocom '98*, 533–539.

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

[24] L. Tassiulas, A. Ephremides (1993). Dynamic server allocation to parallel queues with randomly varying connectivity. *IEEE Trans. Inf. Theory* **39**, 466–478.

[25] X. Wu, R. Srikant (2005). Regulated maximal matching: a distributed scheduling algorithm for multi-hop wireless networks with node-exclusive spectrum sharing. In: *Proc. 44th IEEE CDC-ECC*.

[26] X. Wu, R. Srikant (2006). Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks. In: *Proc. Infocom 2006*.