

# Cross-Layer Scheduling with Infrequent Channel and Queue Measurements

C. Manikandan, Srikrishna Bhashyam, and Rajesh Sundaresan

**Abstract**—The downlink scheduling problem in multi-queue multi-server systems under channel uncertainty is considered. Two policies that make allocations based on predicted channel states are proposed. The first is an extension of the well-known dynamic backpressure policy to the uncertain channel case. The second is a variant that improves delay performance under light loads. The stability region of the system is characterised and the first policy is argued to be throughput optimal. A recently proposed policy of Kar et al [1] has lesser complexity, but is shown to be throughput suboptimal. Further, simulations demonstrate better delay and backlog properties for both our policies at light loads.

**Index Terms**—Backpressure policy, CDMA, cross-layer scheduling, max-weight scheduling, OFDM, polymatching, throughput optimal policy, stability region.

## I. INTRODUCTION

WE study resource allocation and stability in multi-queue multi-server systems under channel uncertainty. The multi-queue multi-server model can be used for downlink of packet data systems based on orthogonal frequency division multiplexing (OFDM) or code division multiple access (CDMA). In an OFDM-based system, each subcarrier or a group of subcarriers can be modeled as a server. In a CDMA-based system, each spreading code can be modeled as a server. Therefore, the code allocation and subcarrier allocation problems in CDMA and OFDM are special cases of the problem considered here. Scheduling decisions are based on available channel and queue state information. Success or failure of a transmission is known only upon explicit feedback from the receiver. Similarly channel state information in frequency-division duplexed systems is known only upon explicit feedback from the receiver. The feedback data is usually received after some delay. To model the delays, we assume that channel states and the results of transmissions are known only once every  $T$  slots, a model used by Kar et al [1], [2]. In this paper, we characterise the so-called stability region for a system with such imperfect channel and queue state information. We also propose a throughput optimal policy and a variant that is aimed

at performing better at light loads. Our results may be easily extended to the uplink.

Andrews et al [3], Kumaran & Viswanathan [5], Agarwal et al [6], have proposed resource allocation policies for 1xEV-DO Rev 0 [4] and other CDMA-based downlinks. Kittipiyakul & Javidi [7] proposed an optimal server allocation policy to minimise average delay under time-varying on-off connectivities, and proposed a heuristic extension to the more general connectivity case in [8]. Mohanram & Bhashyam [9] considered joint power and server allocation to maximise throughput. Tassiulas & Ephremides [10] characterised the stability region of a network and proposed a policy that would stabilise the queues, if at all it was possible to stabilise them. Such policies are called throughput optimal policies. Their policy did not depend on knowledge of arrival rates, and roughly speaking routed traffic from the longest queue to the shortest queue among connected links. The connections were either on or off and were independent and identically distributed (iid) from slot to slot. In the presence of channel uncertainty, the probability of a connection was factored into the weights. The states of all links in a slot were made available to the scheduler at the end of the slot. These results were extended by Neely et al [12] and Neely [13] to a wider class of networks with general loss models, Markov channels, and power control. See also the monograph on this topic by Georgiadis et al [11]. Kar et al [1] recently considered a practical setting where channel state information is available only once every  $T$  slots. They proposed a policy based on virtual queueing, and showed that within such a framework, if there is some policy that will stabilise a set of arrival rates, then so will their policy. The virtual queueing enables easy computation of the best virtual-queue-based schedule.

In this paper, we show that the system model of Kar et al [1] for a multi-queue multi-server system with uncertain connectivity fits within the framework of Georgiadis et al [11] by viewing blocks of  $T$  slots (frame) as a unit for making decisions. The dynamic backpressure policy described in Georgiadis et al [11, p.48] then yields a throughput optimal policy under channel uncertainty as well (Policy 1 below). This policy may also be thought of as an extension of the policy of Kittipiyakul & Javidi [8] for the uncertain channel case. We then argue via an example that the virtual-queue-based policy of Kar et al may have a strictly smaller stability region than that of Policy 1. A simple modification of Policy 1 improves backlog and delay performance at light loads. The complexity of Policy 1 is that of Kar et al scaled by the frame length. The modification requires even more computations.

The rest of the paper is organised as follows. Section II describes the system model and Section III the proposed and

Manuscript received September 24, 2008; revised April 27, 2009; accepted August 14, 2009. The associate editor coordinating the review of this letter and approving it for publication was H. Zheng.

C. Manikandan was with the Department of Electrical Engineering, IIT Madras, Chennai, India. He is currently with Atheros Communications, Chennai (e-mail: mani2004c@gmail.com).

S. Bhashyam is with the Department of Electrical Engineering, IIT Madras, India (e-mail: krishna@ee.iitm.ac.in).

R. Sundaresan is with the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore, India (e-mail: rajeshs@ece.iisc.ernet.in).

This work was supported by Department of Science & Technology, Government of India. This work was presented in part at the National Conference on Communications (NCC 2009) held at IIT Guwahati, India.

Digital Object Identifier 10.1109/TWC.2009.12.081274

other policies. Section IV presents the stability analysis for the system, Section V presents simulation results, and Section VI our conclusions.

## II. SYSTEM MODEL

Consider a downlink system of  $N$  users with one queue each. There are  $K$  servers<sup>1</sup> serving these queues. Transmission is slotted. In each slot, the scheduler decides how servers are allocated across users. Users then transmit their packets on the assigned servers. The effect of the fading channel is modeled by the physical layer's ability to transmit a certain number of packets from the set  $\mathbb{C} = \{0, 1, \dots, c_{\max}\}$ . As each user may see a different channel on each server (as in the OFDM case), the instantaneous channel state at slot  $t$  may be modeled as  $C(t) \in M^{N \times K}(\mathbb{C})$ , a matrix of size  $N \times K$  with entries from  $\mathbb{C}$ . An entry  $C_{nk}(t)$  denotes the number of packets user  $n$  can transmit on server  $k$  at slot  $t$ . For simplicity  $\mathbb{C}$  is the same for all user-server pairs. The process  $(C(t) : t = 0, 1, \dots)$  is taken to be an irreducible (finite-state) Markov chain.

Let  $b_n(t)$  denote the queue size of user  $n$  at slot  $t$ ,  $b(t) = (b_1(t), \dots, b_N(t))^\dagger$  the queue state vector at slot  $t$ , and  $\sum_{n=1}^N b_n(t)$  the backlog at slot  $t$ .

Scheduling is done based on available channel and queue state information. Success or failure of a transmission is known only upon explicit feedback from the receiver. Similarly channel state information in frequency-division duplexed systems is known only upon explicit feedback from the receiver. These are received with delays. To model the delays, we assume that channel states and the results of transmissions are known only once every  $T$  slots, which we call a frame. (The same model was used by Kar et al [1]). More precisely, let the  $(l-1)$ st frame be made of slots  $(l-1)T, (l-1)T+1, \dots, lT-1$ . Scheduling decisions for the  $l$ th frame<sup>2</sup> are made at the start of the frame based on exact queue states and exact channel states for slot  $lT-1$ . Acknowledgements (ACKs) and negative acknowledgements (NACKs) of all the intermediate slots are assumed to be received only at the end of the frame.

In conformance with existing wireless systems, we assume that a server can serve at most one queue in a slot. A queue however may connect to several servers. We may therefore think of a bipartite graph with queues on the left side and servers on the right with connections only between queues and servers, and the degree of any server being at most 1. A set of connections meeting these constraints is called polymatching [1]. The scheduler further decides  $R_{nk}(t)$ , the number of packets that flows across the connection  $n$  to  $k$ . If  $R_{nk}(t) \leq C_{nk}(t)$ , the connection capacity, all packets are received correctly. Otherwise the entire transmission fails. Thus the number of received packets is  $R_{nk}(t)1\{R_{nk}(t) \leq C_{nk}(t)\}$ , where  $1\{\cdot\}$  is the indicator function of an event. This

<sup>1</sup>In an OFDM-based system, a server may refer to each subcarrier or a group of subcarriers. In a CDMA-based system, a server may refer to each spreading code. A server is thus a minimal physical layer resource that is allocable a queue.

<sup>2</sup>Analogous to algorithm FRAME in [12], scheduling, routing and power control are done every slot and queue information is updated once every frame. However, the decisions are based (among other things) on channel information made available only once every frame, unlike algorithm FRAME of [12].

*all-or-none* loss model is motivated by systems that encounter outage when transmitted data rate is higher than the unknown instantaneous link capacity for the slot. It is different from the *optimistic* model of Kar et al [1] where  $\min\{R_{nk}(t), C_{nk}(t)\}$  is received.

## III. SCHEDULING POLICIES

We now present our policies for allocation under channel uncertainty. For slot  $lT+m$  in the  $l$ th frame,  $0 \leq m < T$ , define

$$\tilde{C}_{nk}(lT+m) = \max_r \Pr\{r \leq C_{nk}(lT+m) \mid C_{nk}(lT-1)\}, \quad (1)$$

and  $R_{nk}(lT+m)$  to be the argument that achieves the maximum.  $\tilde{C}_{nk}(lT+m)$  is the maximum expected throughput for user  $n$  on server  $k$  in slot  $(lT+m)$ , given  $C_{nk}(lT-1)$ .

### Policy 1:

- 1) Assign  $w_n \leftarrow b_n(lT-1)$  for  $1 \leq n \leq N$ .
- 2) Repeat the following for each slot  $m$  in the  $l$ th frame,  $0 \leq m < T$ .
  - a) Form the complete bipartite graph where every queue is connected to every server.
  - b) Initialise  $X = \{1, 2, \dots, K\}$ , the set of unallocated servers
  - c) While  $X \neq \emptyset$ 
    - i) Assign  $(n^*, k^*) \leftarrow \arg \max_{n,k} w_n \tilde{C}_{nk}(lT+m)$ .
    - ii) Skip. (Policy 2 is different in this step.)
    - iii) Choose the connection  $(n^*, k^*)$  for the polymatching and let  $R_{n^*k^*}(lT+m)$  packets (see (1)) be transmitted in this slot<sup>3</sup>.
    - iv) Packets may be retransmitted. Packets are chosen according to the lexicographical order based on the pair  $(v, s)$  that is maintained for each packet, where  $v$  is the number of times a packet was transmitted and  $s$  is the sequence number<sup>4</sup>. Increment  $v$  for chosen packets.
    - v)  $X \leftarrow X - \{k^*\}$ .

- 3) Update queue states and channel states based on information from the receivers at the last slot of the frame and reset  $v$  to 0 for all the packets at the start of every frame.

*Remark:* In Step 2.c), the weights  $w_n$  do not change. The search for queue-server connections separates into  $K$  independent searches, one for each server, i.e., to each server  $k$ , connect the queue  $n^*(k) \leftarrow \arg \max_n w_n \tilde{C}_{nk}(lT+m)$ . The complexity is  $O(TKN)$  operations.

**Policy 2:** This policy is the same as Policy 1, except for the following insertion:

- **Step 2.c.ii):** Update  $w_{n^*} \leftarrow [w_{n^*} - \tilde{C}_{n^*k^*}(lT+m)]_+$ , where  $[x]_+ = \max\{x, 0\}$ .

If all  $w_n$  are zero, then reset  $w_n \leftarrow b_n(lT-1)$  for  $1 \leq n \leq N$ .

<sup>3</sup>The queue that has the best queue-size weighted throughput on the server is chosen.

<sup>4</sup>Preference is thus given to packets transmitted the fewest number of times  $v$ , and amongst those transmitted the same number of times, to one with the smallest  $s$ , the earliest to arrive. Retransmissions may occur on different servers.

The motivation for these changes is that weights should adapt to decisions taken, based on expected number of packets that go through. When all weights become zero, all packets have been transmitted roughly equal times, and we may resume retransmissions with original weights. This policy requires  $O((K^2 + KN)T)$  operations.

**The Kar-Luo-Sarkar (KLS) policy** [1]: Maintain  $K$  virtual queues for a user, one associated to each server, in addition to the input queue. In all, there are  $NK + N$  queues. Arrivals during a frame are held at the input queue and allowed to enter the virtual queues only at the start of the frame.  $Q_{nk}$  denotes size of that virtual queue of user  $n$  associated with server  $k$ .

- 1) Queueing: At the start of  $l$ th frame, all input-queued packets of user  $n$  will enter this user's queue associated to server  $k$  if  $k = \arg \min_{k'} Q_{nk'}(lT - 1)$ .
- 2) Service: Compute for every queue-server pair, the weight given by

$$\hat{C}_{nk}(lT) = \frac{1}{T} \mathbb{E} \left[ \sum_{t=lT}^{(l+1)T-1} C_{nk}(t) \middle| C_{nk}(lT - 1) \right].$$

- 3) To server  $k$ , assign the virtual queue  $n^*(k) \leftarrow \arg \max_n Q_{nk}(lT - 1) \hat{C}_{nk}(lT)$ .
- 4) Transmit  $R_{n^*(k)k}(lT + m)$  packets, the argument that maximises (1), in slot  $(lT + m)$ .

*Remarks:* The KLS policy requires  $O(KN)$  operations. The KLS policy fixes the connections for the entire frame while our policies adapt connections to changes in estimated queue sizes and predicted channel conditions within a frame. Note that Step 2 computes expected throughput based on the optimistic loss model, while the transmission decision in Step 4 is based on the all-or-none loss model. Changing either step to be consistent with the other still renders the KLS policy suboptimal because it does not adapt connections within a frame.

#### IV. STABILITY CONSIDERATIONS

1) *Stability region:* A queue is called *strongly stable* if  $\limsup_{L \rightarrow \infty} L^{-1} \sum_{t=0}^{L-1} \mathbb{E}[Q(t)] < \infty$ . A network of queues is strongly stable if all individual queues in the network are strongly stable ([11, Defn. 3.1-3.2]). The *stability region*  $\Lambda$  is the closure of the set of all arrival rates  $\lambda = (\lambda_1, \dots, \lambda_N)^\dagger$  such that there exists a policy (of routing, scheduling, and resource allocation with perfect knowledge of future events) that makes the network strongly stable ([11, Defn. 3.7]).

Let  $\mathbf{C}(l) = (C(lT + m), 0 \leq m < T)$ . We assume that the process  $(\mathbf{C}(l), l = 0, 1, \dots)$  is an irreducible (finite-state) Markov chain with stationary distribution  $P_C$ , and so we may apply the result [11, Th. 3.8] to characterise the stability region. We proceed to do this now.

Let  $r_{nk}^{(m)} \in \mathbb{C}$  packets be scheduled on link  $(n, k)$  in the  $m$ th slot of the  $l$ th frame. The expected throughput on this link, given  $C(lT - 1) = c$ , is

$$\begin{aligned} & \mathbb{E} \left[ r_{nk}^{(m)} \mathbf{1}_{\{r_{nk}^{(m)} \leq C_{nk}(lT + m)\}} \middle| C(lT - 1) = c \right] \\ & = r_{nk}^{(m)} \Pr \left\{ r_{nk}^{(m)} \leq C_{nk}(lT + m) \middle| C(lT - 1) = c \right\}. \quad (2) \end{aligned}$$

The slot  $m$  routing and scheduling matrix of packet flows over all links  $r^{(m)} = (r_{nk}^{(m)}, 1 \leq n \leq N, 1 \leq k \leq K)$  should satisfy the condition that a server serves at most one queue, i.e., each column of  $r^{(m)}$  has at most one nonzero entry. Let  $r = (r^{(m)}, 0 \leq m < T)$  be the routing and scheduling matrices for all the  $T$  slots in the frame. Let  $\mathbb{H}$  be the set of all  $r$  that satisfy the one-queue-per-server condition in every slot. For a choice of  $r \in \mathbb{H}$ , let

$$g_{nk}(r, c) = \sum_{m=1}^T r_{nk}^{(m)} \Pr \left\{ r_{nk}^{(m)} \leq C(lT + m) \middle| C(lT - 1) = c \right\}$$

be the conditional expected throughput on link  $(n, k)$  when queue  $n$  is sufficiently loaded. Let  $g(r, c) = (g_{nk}(r, c), 1 \leq n \leq N, 1 \leq k \leq K)$  be the matrix of conditional expected throughputs on each link. Let  $R$  be an  $\mathbb{H}$ -valued random variable with conditional distribution  $P_{R|C}(\cdot|c)$  and with the interpretation that policy  $r \in \mathbb{H}$  is picked with probability  $P_{R|C}(r|c)$  when  $C(lT - 1) = c$ . Thus  $P_{R|C}$  represents a randomised policy. Let  $G = \mathbb{E}[g(R, C)]/T$  be the per-slot expected throughput matrix under full load for the randomised policy  $P_{R|C}$ , averaged over the channel realisations. Finally, let  $\Gamma$  be the closure of the set of all such  $G$ .

From [11, Th. 3.8] the stability region  $\Lambda$  is the set of all  $\lambda \in \mathbb{R}_+^N$  such that there is a  $G \in \Gamma$  and a flow  $(f_{nk} : 1 \leq n \leq N, 1 \leq k \leq K)$  satisfying (a)  $0 \leq f_{nk} \leq G_{nk}$  for every  $(n, k)$ , and (b)  $\lambda_n = \sum_{k=1}^K f_{nk}$  for every  $n$ . This completely characterises the stability region.

2) *Throughput optimality of Policy 1:* Let  $\text{int}(\Lambda)$  denote the interior points of the stability region. From the definition of the stability region, if  $\lambda \in \text{int}(\Lambda)$ , there exists a policy that makes the network with arrival rates  $\lambda$  strongly stable. However, [11, Th. 4.5] states the stronger result that every  $\lambda \in \text{int}(\Lambda)$  is strongly stabilised by a single policy, the *dynamic backpressure policy*, that does not depend on  $\lambda$ . A policy with this property is called *throughput optimal*. It is straightforward to verify that the dynamic backpressure policy specialised to our system is indeed Policy 1. Consequently, Policy 1 is throughput optimal.

If the arrivals are iid from slot to slot, the notion of stability can be strengthened to one of positive recurrence of the nontransient queue states as studied by Tassiulas & Ephremides [10] and Kar et al [1]. See Neely et al [12] and Georgiadis et al [11, pp.53-54] for this strengthening.

Note that Policy 2 is a heuristic modification of Policy 1 to achieve better delay performance at low traffic. Simulation results are presented in Section V to show the improvement in performance over Policy 1 using Policy 2. The question of throughput optimality of Policy 2 remains open.

3) *The KLS policy:* We now compare the KLS policy with the dynamic backpressure algorithm. Recall that we have a two-hop network from users to servers, where user  $n$  with an input queue is connected to  $K$  virtual queues,  $q_{nk}, 1 \leq k \leq K$ , and  $q_{nk}$  is connected only to server  $k$  and no other server. The link from the input queue to each virtual queue is noiseless; the throughput is just the number of packets flowing through this link. The dynamic backpressure algorithm picks the virtual queue  $k$  that maximises  $(b_n(lT - 1) - Q_{nk}(lT - 1))b_n(lT - 1)$  for each  $n$ . Clearly the  $k$  with the least  $Q_{nk}(lT - 1)$  is picked, as done by the KLS policy.

Next, server  $k$  should be connected to at most one of the virtual queues  $q_{1k}, \dots, q_{Nk}$ . The dynamic backpressure algorithm picks that virtual queue  $n^*(k, m)$  in slot  $m$  that maximises the weighted expected throughput  $Q_{nk}(lT - 1)\check{C}_{nk}(lT + m)$  under the all-or-none loss model. (Alternatively, it picks  $n^*(k, m)$  that maximises  $Q_{nk}(lT - 1)\mathbb{E}[C_{nk}(lT + m)|C(lT - 1)]$  under optimistic channel model). Thus, the dynamic backpressure algorithm adapts to changing statistics within the frame. However, the KLS policy retains a single queue-server connection through all slots in the frame, chosen to maximise weighted total expected throughput in the frame. It will have a smaller stability region than Policy 1 as illustrated in the fast fading example below. For slow fading, the stability regions for Policy 1 and KLS Policy are not significantly different (See section V). Consider the probability transition matrices given below.

$$\begin{bmatrix} \delta & 1 - \delta \\ 1 - \delta & \delta \end{bmatrix}, \begin{bmatrix} \delta & \frac{1 - \delta}{3} & \frac{1 - \delta}{3} & \frac{1 - \delta}{3} \\ \frac{1 - \delta}{3} & \delta & \frac{1 - \delta}{3} & \frac{1 - \delta}{3} \\ \frac{1 - \delta}{3} & \frac{1 - \delta}{3} & \delta & \frac{1 - \delta}{3} \\ \frac{1 - \delta}{3} & \frac{1 - \delta}{3} & \frac{1 - \delta}{3} & \delta \end{bmatrix}. \quad (3)$$

To model a fast fading channel,  $\delta$  should be small. Let  $T = 2$  and  $c_{\max} = 1$ . Thus, we have only on-off channels; all-or-none and optimistic loss models coincide. Consider two queues ( $N = 2$ ) and one server ( $K = 1$ ) with identical channel transition probabilities given by the left matrix in (3). A static allocation as in the KLS policy will only get the average, whereas an adaptive policy can match to the near cyclic channel variation within a frame. The stationary distribution puts equal mass on all four channel possibilities. A sum rate (over the frame) of nearly 1 packet (in channel state 00 at slot  $lT - 1$ ), 1 packet (channel state 11), 2 (channel state 01), and 2 (channel state 10), yielding total sum rate of nearly  $(1+1+2+2)/4 = 6/8$  packets per slot can be achieved. More precisely, the symmetric rate  $(r(\delta), r(\delta))^\dagger$  with  $r(\delta) = (1+(1-\delta)+(1-\delta)^2+\delta^2)/8$  per slot is achievable, yielding a sum rate  $2r(\delta) \rightarrow 6/8$  as  $\delta \downarrow 0$ . The maximum possible sum rate for the KLS policy is upper bounded by  $(4+2\delta(1-2\delta))/8 < 5/8$  for sufficiently small  $\delta$ . The stability region of Policy 1 is therefore strictly larger than that of the KLS policy. See Figure 1.

The above discussion also shows that the KLS policy is the dynamic backpressure policy under virtual queueing subjected to the constraint that connections cannot change within a frame.

## V. SIMULATION RESULTS

Consider a system with  $N = 128$  users and  $K = 16$  servers. The channel state, in terms of the number of transmissible packets in a slot, is modeled as a Markov chain. This Markov chain is composed of smaller independently evolving and identical Markov chains on four states, one for each user-server pair. So  $\mathbb{C} = \{0, 1, 2, 3\}$ . The transition probabilities are given by the right matrix in (3) with  $\delta = 0.98$ , and the initial distribution is the stationary distribution on each user-server

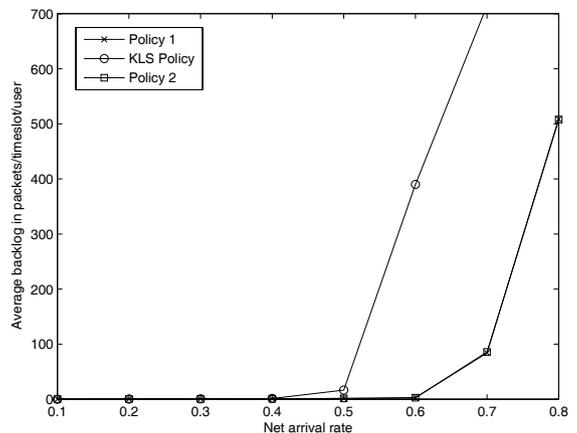


Fig. 1. Average backlog comparison for on-off channel model for  $\delta = 0.1$ . The net arrival rate beyond which backlog rapidly increases signifies the threshold of the stability region.

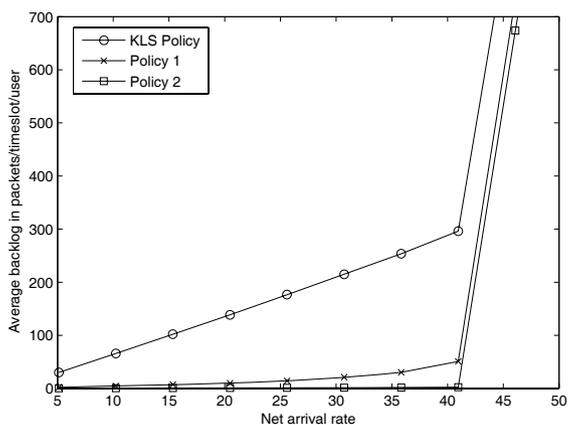


Fig. 2. Average backlog comparison for  $T=8$  for symmetric arrivals at high traffic.

pair. Arrivals to queues are truncated Poisson with a maximum of 100 arrivals. All users have the same mean number of arrivals  $\lambda$  per slot. The abscissa in all plots is the mean total arrivals per slot, summed over all users. Backlog and delay are used as metrics for comparison. Backlog is measured only at frame boundaries.

1) *Average backlog comparison for a fixed  $T$* : Average backlog in packets/time-slot/user is calculated and plotted for all described policies against total arrival rate in Figures 2 and 3.  $T$  is set to 8. From the plots, we infer that all the policies have similar performance at high rates. However, our proposed policies outperform the KLS policy at low rates as can be seen from the magnification in Figure 3. Simulation results for other values of  $T$  and asymmetric arrival rates exhibit the same qualitative behaviour. Given the strongly diagonal nature of the channel transition matrix, the channel remains nearly static across the frame, and it should not be surprising that all policies result in instability at roughly the same threshold sum rate.

2) *Delay comparison for a fixed  $T$* : Figure 4 plots the delay of all transmitted packets. For each policy, the figure contains

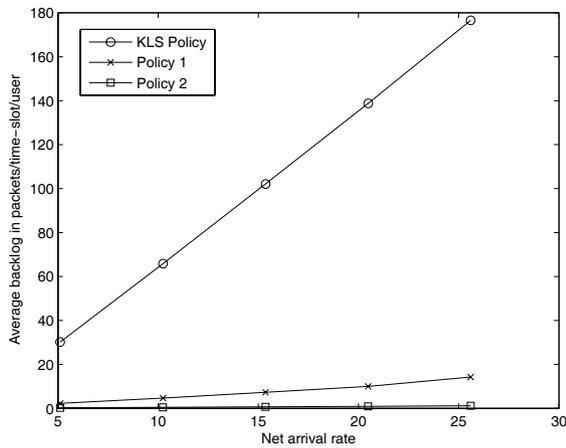


Fig. 3. Average backlog comparison for  $T=8$  for symmetric arrivals at low traffic.

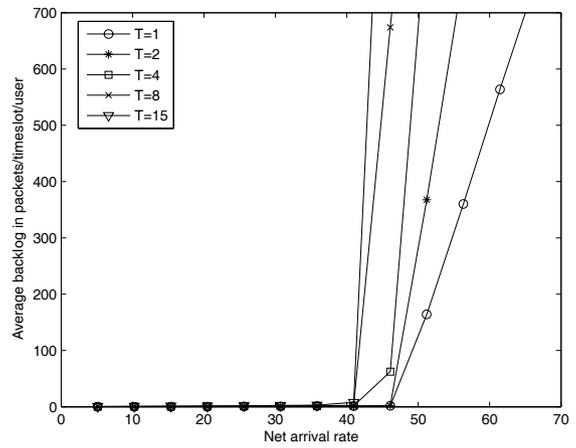


Fig. 5. Average backlog of Policy 2 versus  $T$  for symmetric arrivals.

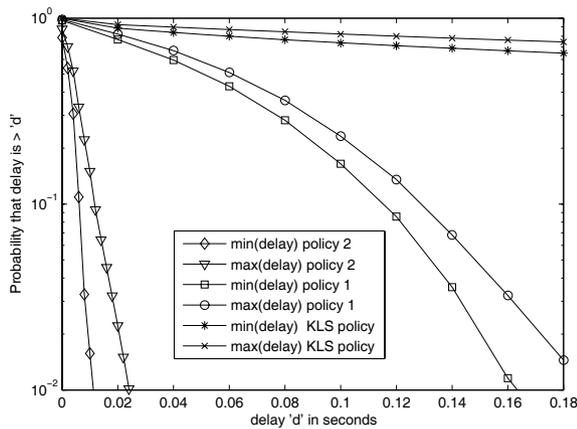


Fig. 4. Delay comparison for symmetric arrivals,  $T=4$ , net arrival rate=25.6.

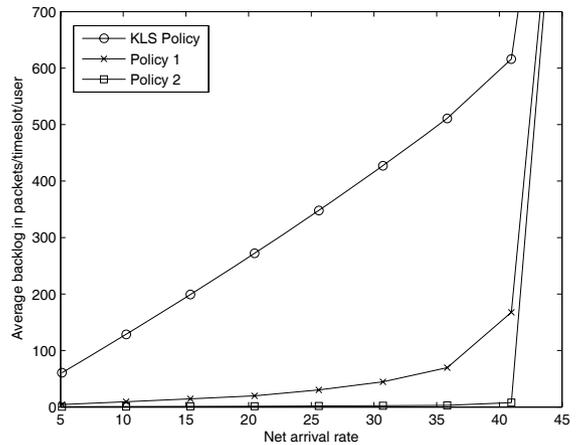


Fig. 6. Average backlog comparison for  $T=15$  for symmetric arrivals at high traffic.

a best and worst case value for the delay distribution at each value of delay. The proposed policies give significantly better delay performance than the KLS policy. The net arrival rate is 25.6. Similar results were obtained for rates in the range 5 to 40.

3) *Average backlog comparison versus  $T$* : Figures 5 plots the backlog of Policy 2 for different  $T$ . It is interesting to note that the stabilisable sum rate remains roughly the same for all the considered  $T$ . This is due to the slow fading nature of the channel and the adaptation using predicted channel states. The transition behaviour from a stable system to unstable queues is different across  $T$ . The degradation in performance with increasing  $T$  is more pronounced in KLS policy than policy 2. This is illustrated in Figure 6 by comparing the two policies for  $T = 15$ . The gap between the two policies for  $T = 15$  is higher than the gap for  $T = 8$  (shown in Figure 2). This is due to the inability of the KLS policy to adapt to the channel variations.

## VI. CONCLUSIONS

We showed that scheduling and allocation for a specific model of channel uncertainty falls within the framework

described in Georgiadis et al [11]. For throughput optimality, it is important to adapt to predicted channel states across slots within a frame. The dynamic backpressure algorithm (Policy 1) does this while the KLS policy does not. Policy 1 is throughput optimal. Moreover, Policies 1 and 2 outperform the KLS policy in the light load regime. Naturally, our policies require more computations. The complexities of the KLS policy, Policy 1, Policy 2 are  $O(KN)$ ,  $O(TKN)$ , and  $O(T(K^2 + KN))$  operations, respectively. Power control based allocation can be easily added to the framework by choosing the power allocation matrix that maximises the queue-weighted sum of rates as in [12]. The proposed policies can be directly applied to CDMA/OFDM based wireless systems. The degradation due to infrequent channel and queue measurements in realistic systems can be minimized by using the proposed policies.

## REFERENCES

[1] K. Kar, X. Luo, and S. Sarkar, "Throughput-optimal scheduling in multi-channel access point networks under infrequent channel measurements," in *Proc. IEEE Infocomm 2007*, May 2007.

- [2] K. Kar, X. Luo, and S. Sarkar, "Throughput-optimal scheduling in multi-channel access point networks under infrequent channel measurements," *IEEE Trans. Wireless Commun.*, vol. 7, no. 7, pp. 2619-2629, July 2008.
- [3] M. Andrews, K. Kumaran, K. Ramanan, A. L. Stolyar, R. Vijayakumar, and P. Whiting, "Providing quality of service over a shared wireless link," *IEEE Commun. Mag.*, vol. 39, no. 2, pp. 150-154, Feb. 2001.
- [4] "TIA/EIA/IS-856 cdma2000 High Rate Packet Data Air Interface Specification," Telecommunications Industry Association. [Online]. Available: [www.3gpp2.org](http://www.3gpp2.org).
- [5] K. Kumaran and H. Viswanathan, "Joint power and bandwidth allocation in downlink transmission," *IEEE Trans. Wireless Commun.*, vol. 4, no. 3, pp. 1008-1016, May 2005.
- [6] R. Agarwal, V. Subramanian, and R. Berry, "Joint scheduling and resource allocation in CDMA systems," *2<sup>nd</sup> Workshop Modeling Optimization Mobile, Adhoc Wireless Networks (WiOPT '04)*, Cambridge, UK, Mar. 2004.
- [7] S. Kittipiyakul and T. Javidi, "Delay-optimal server allocation in multi-queue multi-server systems with time-varying connectivities," *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2319-2333, May 2009.
- [8] S. Kittipiyakul and T. Javidi, "Resource allocation in OFDMA with time-varying channel and bursty arrivals," *IEEE Commun. Lett.*, vol. 11, no. 9, Sep. 2007.
- [9] C. Mohanram and S. Bhashyam, "Joint subcarrier and power allocation in channel-aware queue-aware scheduling for multiuser OFDM," *IEEE Trans. Wireless Commun.*, vol. 6, no. 9, Sep. 2007.
- [10] L. Tassiulas and A. Ephremides, "Stability properties of constrained queuing systems and scheduling for maximum throughput in multihop radio networks," *IEEE Trans. Automatic Control*, vol. 37, no. 12, pp. 1936-1949, 1992.
- [11] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Foundations Trends Networking*, vol. 1, no. 1, pp. 1-144, 2006.
- [12] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic power allocation and routing for time varying wireless networks," *IEEE J. Sel. Areas Commun.*, special issue on wireless ad-hoc networks, vol. 23, no. 1, pp. 89-103, Jan. 2005.
- [13] M. J. Neely, "Dynamic power allocation and routing for satellite and wireless networks with time varying channels," Ph.D. dissertation, Massachusetts Institute of Technology, LIDS, 2003.