

Draining Time based Scheduling Algorithm

Vijay G. Subramanian and Douglas J. Leith
Hamilton Institute, National University of Ireland,
Maynooth, Co. Kildare, Ireland.
Email: [vijay.subramanian,doug.leith]@nuim.ie.

Abstract—Distributed scheduling and routing policies that perform well are increasingly being sought for multi-hop wireless infrastructure networks. The state-of-the-art *max-pressure* algorithms have provably optimal stability properties but in networks with cycles and loops the delay performance can be quite poor at rates that are not very close to the capacity region boundary. We propose a new scheduling and routing policy to try to address this that results in serving the longest queue but routing packets along (dynamic) shortest-path routes. For a two-node relay we prove stability for our policy using fluid limits.

I. INTRODUCTION

Scheduling, resource allocation and routing have traditionally been important topics of study for communication networks. However, with the incredible number of devices of today that can help connect people there has been a surge of activity in this area. Refer to [1], [2] for a current review of work to date and research problems in this area. For our purposes we are interested in multi-hop wireless infrastructure networks, i.e., networks of wireless access points that are fixed and can communicate with each other to transmit data between different sources and destinations. Additionally, to keep the complexity low we restrict our attention to distributed algorithms that could involve (limited) communication between neighbour nodes. The state of the art in this area is to consider *max-pressure* based algorithms [3], [4], [5], [6], [7] that use information about the queue-length of different flows at neighbour nodes. These policies are provably stabilising, i.e., can keep the queues stable as long

as the arrival rates lie within the capacity region of the system. This is usually proved using the sum of squares of the queue-lengths at the different nodes as a Lyapunov function. The policy can usually be written as one-step maximal weight matching procedure, i.e., considering only nearest/one-hop-distant neighbours. This, however, precludes the logic from knowing about loops and avoiding them. Therefore, at rates not close to the capacity boundary, one can demonstrate that packets spend a lot of time in loops before exiting the system, leading to large delays: refer to Section II for an illustrative example. This lack of knowledge of the topology of the network also allows *max-pressure* policies to route packets to nodes that do not communicate with the destination node. This does not impact stability but does adversely impact applications. It is possible that combining network coding with *max-pressure* scheduling [8] the performance issues highlighted above can be solved. However, without network coding we propose a policy that involves constructing distributed metrics that can reveal the minimum “distance” to the destination from each node which we term the “draining time” from each node. This information is used to prune the routes to be chosen. In Section II we describe our model and discuss a model where *max-pressure* performs poorly, and then present our policy in Section III. Finally, in Section IV we prove stability properties for the constructed policy for two-node relay networks. In the interests of brevity we omit some of the detailed proofs. We, however, point the reader to [9] for all details and an extensive bibliography.

Work supported by Science Foundation Ireland grant IN3/03/I346.

II. MODEL AND EXAMPLE

We consider a discrete-time queueing system where packets spend at least one time-slot in each queue, and define the following: let N be the set of nodes in the network and F is the set of flows using the network with source $s(f) \in N$ and destination $d(f) \in N$ for flow $f \in F$. Let $Q_n^f(t)$ be the queue-length of flow f at node n (at time t). Define connectivity matrix $C - N \times N$ with $C_{nm} \in \{0, 1\}$, and rate matrix R that delineates the number of (whole) packets that can be served in each time-unit. We assume that there are no interference constraints, and that different receivers can simultaneously receive transmissions from different transmitters based upon a rate matrix as will be specified in detail. The underlying assumption is of enough frequencies to ensure this in the case for a network of wireless access points, say 802.11e/g access points, and a distributed frequency assignment such as that proposed in [10] to yield a non-interfering allocation. If one models the impact of interference to be such that some packets in transit get dropped with the exact number that are dropped being a function of the scheduled transmissions that interfere, then we believe that the policy presented in this paper can be extended to include interference.

From the definition of the connectivity matrix C we have the following for all $n, m \in N$ $1_{[R_{nm}>0]} = C_{nm}$, where we have 1_S being the indicator function of set S . Using this we assume that each transmission from n of flow f at time t consists of $\min(Q_n^f(t), \max_{m:R_{nm}>0} R_{nm})$ packets and each receiver m with $R_{nm} > 0$ receives the first $\min(Q_n^f(t), R_{nm})$ packets. Note that the model allows for policies more general than just routing but using the results from [3], [4], [6], [7] it is sufficient to consider routing to ensure stability. Define $R_{\min} := \min_{n,m \in N: R_{nm}>0} R_{nm}$ and $R_{\max} = \max_{n,m \in N} R_{nm}$ to be the minimum rate and (finite) maximum rate, respectively. Note that this work concentrates on networks with constant service-rates, and wireless networks never fit this billing. However, given that we are interested in wireless infrastructure networks there exist reasonably long time-scales when such a notion makes

sense. One can extend the policy assuming that the link rates change due to fast-fading but with the connectivity matrix remaining fixed. This is a natural setting for our problem. Thus, we plan on using the version for static link rates as a starting point, and later on generalising to time-varying link capacities.

We first show an example of a network with one flow where *max-pressure* performs poorly. Consider the network shown in Fig. 1. The operation of a typical *max-pressure* algorithm yields the following. At stations 1, 2, 4, route to next station if queue-length of next station is smaller. At station 3, if $Q_3(t) > Q_5(t)$ and $Q_3(t) \leq 2Q_4(t) - Q_5(t)$, then route $\min(1, Q_3(t))$ packets to station 5; and if $Q_3(t) > Q_4(t)$ and $Q_3(t) > 2Q_4(t) - Q_5(t)$, then route $\min(2, Q_3(t))$ packets to station 4. Note that the policy (*MP*) is not work-conserving and can route packets to station 4. We will compare this with a work-conserving policy (*DT*) that avoids sending packets to station 4. For this network our proposal will coincide with *DT*. With *i.i.d.*

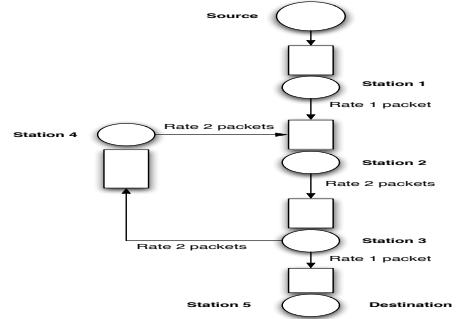


Fig. 1. Example network with loops.

$\text{Geom}(\frac{\lambda}{1+\lambda})$ arrivals we will compare the two policies by simulation. The two algorithms are compared at two arrival rates 0.9 and 0.75, and it can be demonstrated that the delay problems are exacerbated at lower rates. From the results in Tables I and II it is clear that the *MP* algorithm leads to much higher delay and higher delay standard deviation than the *DT* algorithm. In fact, one can also see that the number of visits to the stations ($K(N)$) in the loop actually increases with

the lower arrival rate. Another illustration that the *MP* algorithm can lead to problems with loops is that at rate $\lambda = 0.9$ we have 0.3% of the packets going through station 4, but at rate $\lambda = 0.75$ we have 1.64% going through station 4. From our simulations we have found that with increasing burstiness of the arrival processes the performance of the *MP* algorithm worsens.

TABLE I
SIMULATION RESULTS FOR $\lambda = 0.9$.

Parameter	<i>MP</i> Algorithm	<i>DT</i> Algorithm
Mean delay	22.953	12.165
Std. dev. delay	89.001	8.284
$E[Q_1]$	11.279	8.133
$E[Q_2]$	3.431	0.892
$E[Q_3]$	2.41	0.934
$E[Q_4]$	2.78	0.0
$\mu(K(2))$	1.185	1
$\sigma(K(2))$	6.011	0
$\mu(K(4))$	0.185	0
$\sigma(K(4))$	6.011	0

TABLE II
SIMULATION RESULTS FOR $\lambda = 0.75$.

Parameter	<i>MP</i> Algorithm	<i>DT</i> Algorithm
Mean delay	19.835	7.316
Std. dev. delay	58.518	3.952
$E[Q_1]$	6.2	3.254
$E[Q_2]$	3.286	0.754
$E[Q_3]$	2.256	0.754
$E[Q_4]$	2.5	0.0
$\mu(K(2))$	1.479	1
$\sigma(K(2))$	7.362	0
$\mu(K(4))$	0.479	0
$\sigma(K(4))$	7.362	0

III. SCHEDULING ALGORITHM

For each flow f at time t define the metric $V_t^f : N \rightarrow \mathbb{R}_+$ as follows. For the destination nodes set $V_t^f(d(f)) = 0$. For all $n \in N \setminus \{d(f)\}$ if n communicates with $d(f)$ set $V_t^f(n) = \min_{m \in N} \frac{Q_n^f(t)}{R_{nm}} + V_t^f(m)$. Note that node n communicates with node m if $\exists k \in \mathbb{N}$ such that $C_{nm}^k > 0$ where C^k is the k^{th} power of the connectivity matrix C . For all $n \in N \setminus \{d(f)\}$ if n does not communicate with

$d(f)$ set $V_t^f(n) = +\infty$. We have the following inequalities $\frac{Q_n^f(t)}{R_{\max}} \leq V_t^f(n)$, and if $V_t^f(n) < +\infty$, then $V_t^f(n) \leq \frac{\sum_{n \in N} Q_n^f(t)}{R_{\min}}$. Thus, the metric $V_t^f(n)$ is roughly linear in the queue-length, and is a surrogate to the draining time or delay to destination. This is used since calculating the true draining time (related to make-span) or delay-to-destination is computationally prohibitive [11], [12]. Note that $V_t^f(i)$ can also be written as the minimum of the sum of $\frac{Q_n^f(t)}{R_{nm}}$ along all paths from i to $d(f)$. Now the distributed Bellman-Ford algorithm [13] yields $V_t^f(n)$; the computational complexity of which is proportional to the number of links of the network. Note that we can ignore all paths with loops from i to $d(f)$ since dropping the loop will only reduce the path cost. In practice we could augment the link metrics to $a_{n,m} + \frac{b_n^f Q_n^f(t)}{R_{nm}}$ for some $a_{n,m} > 0$ and $b_n^f > 0$ to provide QoS differentiation, and use virtual queues instead of $Q_n^f(t)$ to control the delays (the latter at a small loss in capacity). For implementation purposes one can also perform the computation of the V metrics slower than once every scheduling instance.

At every node n define the following metric for every flow f and node m given by

$$W_{n,m}^f(t) := R_{nm}(V_t^f(n) - V_t^f(m)) \text{ if } V_t^f(n) < +\infty, \quad (1)$$

and $W_{n,m}^f(t) := -\infty$ otherwise. Use this to define a flow metric $F_n^f(t)$ at node n given by $F_n^f(t) := \max_{m \in N} W_{n,m}^f(t)$ with the maximiser given by $m_{f,n}^*(t)$. Now define a node metric $G_n(t) := \max_{f \in F} F_n^f(t)$ with the maximiser given by $f_n^*(t)$. Note that in contrast to the *max-pressure* algorithm we are replacing $Q_n^f(t)$ by $V_n^f(t)$. Ties are broken in an arbitrary manner in both maximisations.

Scheduling algorithm: If $G_n(t) > 0$ serve flow $f_n^*(t)$ and route $\min(Q_n^{f_n^*(t)}(t), R_{nm_{f_n^*(t),n}})$ packets to $m_{f_n^*(t),n}^*(t)$.

Alternate view: Restricting attention only to nodes that communicate with $d(f)$ the definition of $V_t^f(m)$ leads to the following relation for all m , $R_{nm}(V_t^f(n) - V_t^f(m)) \leq Q_n^f(t)$ with equality for some m with $C_{nm} = 1$; this is specifically given by an m such that $P_t^f(n, m) = 1$ where one

defines the $N \times N$ matrix P_t^f such that $P_t^f(n, m) = 1$ if $V_t^f(n) = \frac{Q_n^f(t)}{R_{nm}} + V_t^f(m)$ and $V_t^f(n) < +\infty$ and $P_t^f(n, m) = 0$ otherwise. Therefore, $F_n^f(t) = Q_n^f(t)$ and $G_n(t) = \max_{f \in \mathcal{F}} Q_n^f(t)$. This implies that $f_n^*(t) \in \arg \max_{f \in \mathcal{F}: V_n^f(t) < +\infty} Q_n^f(t)$ and $m_{f_n^*(t), n}^*(t) \in \left\{m \in N : P_t^{f_n^*(t)}(n, m) = 1\right\}$. In other words, the longest queue is served at each node, and routing is along (dynamic) shortest paths (to destination $d(f_n^*(t))$) with link metric $\frac{Q_n^f(t)}{R_{nm}}$. It is well documented that *max-pressure* policies are not work-conserving. In fact, it is clear that a node does not serve any flow if locally (over all neighbour nodes) it has the smallest vector of queue-lengths. Note from the alternate definition that, in contrast, our policy is work-conserving.

Queueing equation: Let $\beta_{n,m}^f(t) \in \{0, 1\}$ be the routing variables, i.e., the indicator function of whether flow f packets are routed from node n to node m . Note that by conservation of time we have $\sum_{f \in \mathcal{F}} \sum_{m \in N} \beta_{n,m}^f(t) \leq 1$. If the policy is work-conserving, then we have equality if there is any flow to be served, i.e., if $\max_f Q_n^f(t) > 0$. The queueing equations for flow f at node n are given as follows. If $n \neq d(f)$, then

$$Q_n^f(t+1) - Q_n^f(t) = \sum_{m \in N} \beta_{m,n}^f(t) \min(R_{mn}, Q_m^f(t)) + A_n^f(t) - \sum_{m \in N} \beta_{n,m}^f(t) \min(R_{nm}, Q_n^f(t)), \quad (2)$$

where the first term is due to services at other nodes in the network, the second term from new arrivals at node n , and the final term owing to services at node n . If $n = d(f)$, then one can drop the terms for the queue-length at time t and the service at node n because the destination node for each flow absorbs all packets that were present in the past time-slot.

We use fluid limits [14], [15] for our proofs. For this purpose we will be considering a sequence of networks with the same topology, arrival processes and service rates but with increasing scaling parameter $M = \sum_{f \in \mathcal{F}} \sum_{n \in N} Q_n^f(0)$. We assume that the arrival processes satisfy a Functional Strong Law of Large Numbers with mean rate λ_n^f - independent arrival processes with countable state Markov Chains will satisfy this: also refer to [14], [15] for other processes that will satisfy these

conditions.

For a flow f such that node n does not communicate with $d(f)$ as given by the connectivity matrix C , we set $Q_n^f(0) = 0$ in all the networks for consideration. Our policy never routes flow f to such nodes. However, *max-pressure* policies can route packets to such nodes, *albeit* only a finite number of times. We extend our discrete-time processes, namely $\{Q_n^f(t) : \forall n \in N, \forall f \in \mathcal{F}, \text{ and } \forall t = 0, 1, \dots\}$ to continuous time by setting $Q_n^f(t) = Q_n^f([t])$ resulting in right-continuous processes with left-hand limits on $[0, \infty)$, i.e., $\mathbb{D} := \mathbb{D}[\mathfrak{R}_+; \mathfrak{R}^k]$, $k \in \mathbb{N}$ that we endow with the Skorohod J_1 -topology. Convergence in \mathbb{D} to continuous functions \mathbb{C} is given by *uniform convergence on compact sets (u.o.c.)*.

For every n and f we will look at the behaviour of $Q_n^{f,M}(t) := \frac{Q_n^f(Mt)}{M}$ for all $t \in \mathfrak{R}_+$ in the limit as $M \rightarrow +\infty$. For node $n = d(f)$ we can assert [9] that $q_n^f(t) = \lim_{M \rightarrow \infty} Q_n^{f,M}(t) = 0$ a.s. u.o.c for all $t > 0$. Therefore, we need not be concerned about the destination nodes for stability issues. For a regular node we have

$$Q_n^f(t) \leq Q_n^f(0) + [t]NR_{\max} + \sum_{s=0}^{[t]-1} A_n^f(s). \quad (3)$$

This will imply uniform integrability of $Q_n^{f,M}(t)$ from assumptions about the arrival processes [14], [15]. Therefore demonstrating the stability of the fluid limit implies stability of the original queueing network. We have [9] that along a sub-sequence $\lim_{M \rightarrow \infty} Q_n^{f,M}(t) = q_n^f(t)$ u.o.c. where $q_n^f(t)$ is non-negative and Lipschitz continuous. Moreover, we have the relationship a.s.

$$q_n^f(t) = q_n^f(0) + \lambda_n^f t - \sum_{m \in N} \sum_{k=1}^{R_{nm}} k \phi_{n,m}^{f,k}(t) + \sum_{m \in N} \sum_{k=1}^{R_{mn}} k \phi_{m,n}^{f,k}(t), \quad (4)$$

where $\phi_{n,m}^{f,k}(t)$ is a non-decreasing Lipschitz continuous function that measures the amount of time in $[0, t]$ that node n routes $k \in \{0, 1, \dots, R_{nm}\}$ packets of flow f to node m . For all policies if $q_n^f(t) = 0$, then $\sum_{k=1}^{R_{nm}} \frac{d\phi_{n,m}^{f,k}}{dt}(t) = 0$ for all $m \in N \setminus \{n\}$.

For every t define function $v_t^f(n)$ in a similar manner to $V_t^f(n)$ as follows. If $n = d(f)$, then set $v_t^f(n) = 0$. If node $n \neq d(f)$ communicates with $d(f)$, then set $v_t^f(n) = \min_m \frac{q_n^f(t)}{R_{f,m}} + v_t^f(m)$. Otherwise for nodes $n \neq d(f)$ set $v_t^f(n) = +\infty$. Also define the $N \times N$ matrix p_t^f such that $p_t^f(n, m) = 1$ if $v_t^f(n) = \frac{q_n^f(t)}{R_{n,m}} + v_t^f(m)$ and $v_t^f(m) < +\infty$ and $p_t^f(n, m) = 0$ otherwise. One can show the convergence of the scaled V metrics to $v_t^f(n)$. Then our policy has the following properties [9]:

Maximum Rate Service: If $q_n^f(t) > 0$, $t \in [t_1, t_2]$ for every f and m , then $\sum_{k=1}^{R_{nm}-1} \frac{d\phi_{n,m}^{f,k}}{dt}(t) = 0$ and only $k = 0$ or $k = R_{nm}$ can be chosen.

Scheduling: Only the longest queues get served, i.e., if $\hat{f} \notin \arg \max_{f \in F} q_n^f(t)$, then $\sum_{m \in N} \sum_{k=1}^{R_{nm}} \frac{d\phi_{n,m}^{\hat{f},k}}{dt}(t) = 0$. The allocation between flows with the longest queues is arbitrary, and determines the switching curves between flows at any given node.

Work Conserving: If $\max_f q_n^f(t) > 0$, then $\sum_{\hat{f} \in \arg \max_{f \in F} q_n^f(t)} \frac{d\phi_{n,m}^{\hat{f},R_{nm}}}{dt}(t) = 1$.

Routing: Only the shortest paths are taken for routing, i.e., if $R_{n,\hat{m}}(v_t^{\hat{f}}(n) - v_t^{\hat{f}}(\hat{m})) < \max_f q_n^f(t)$ for some \hat{f} and \hat{m} , then $\sum_{k=1}^{R_{n,\hat{m}}} \frac{d\phi_{n,\hat{m}}^{\hat{f},k}}{dt}(t) = 0$.

We can summarise all of the above by: if $\max_f q_n^f(t) > 0$, then set $\sum_{\hat{f} \in \arg \max_{f \in F} q_n^f(t)} \sum_{m: p_{\hat{f}}^{\hat{f}}(n, m)=1} \frac{d\phi_{n,m}^{\hat{f},R_{nm}}}{dt}(t) = 1$. The allocation between neighbour nodes when two or more nodes m have $p_{\hat{f}}^{\hat{f}}(n, m) = 1$ is arbitrary, and determines the switching curves between routes at any given node.

Using the characterisation in [3], [4], [5], [6], [7] given a set of arrival rates $\{\lambda_n^f\}_{(f \in F, n \in N)}$ with $\lambda_n^f = 0$ if n does not communicate with $d(f)$, for stability it is sufficient that there exist non-negative numbers $\{\psi_{nm}^f\}_{(f \in F, n \in N, m \in N)}$ such that the following hold:

Feasibility: If $R_{nm} = 0$, then $\sum_f \psi_{nm}^f = 0$.

Time Conservation: $\sum_f \sum_{m \in N} \psi_{nm}^f \leq 1$.

Flow Conservation: We have $\lambda_n^f +$

$\sum_{m \in N} R_{mn} \psi_{mn}^f < \sum_{m \in N} R_{nm} \psi_{nm}^f$. An equality here is necessary for stability.

IV. SIMPLE EXAMPLE: A TWO-NODE RELAY

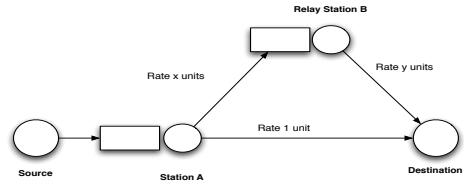


Fig. 2. Two-node Relay.

Consider the relay network shown in Fig. 2 obtained by scaling an original model with integral rates. We assume $x > 1$, otherwise our algorithm is trivially stabilising since all the traffic will be routed directly to the destination. Then stability condition is $\lambda < \min(x, 1 + \frac{x-1}{x}y)$. For this network the only decision variable is at node A : to route traffic to node B or to the destination directly. Our routing policy is the following:

$$r(t) = \begin{cases} D & \text{if } q_A(t) \frac{x-1}{x}y \leq q_B(t); \\ B & \text{otherwise.} \end{cases} \quad (5)$$

We have chosen to route to the destination directly when both routing decisions are equally good.

We will show that the fluid limit is stable. For the fluid limit we have the following:

$$\begin{aligned} \frac{dq_A(t)}{dt} &= \begin{cases} \lambda - 1_{[q_A(t) > 0]} & \text{if } r(t)=D; \\ \lambda - x 1_{[q_A(t) > 0]} & \text{if } r(t)=B; \end{cases} \\ \frac{dq_B(t)}{dt} &= -y 1_{[q_B(t) > 0]} + \begin{cases} 0 & \text{if } r(t)=D \\ x 1_{[q_A(t) > 0]} & \text{if } r(t)=B \end{cases}. \end{aligned} \quad (6)$$

Consider quadratic Lyapunov function $L(t) = \frac{q_A^2(t) + \frac{q_B^2(t)}{y}}{2}$. Then we have the following:

$$\begin{aligned} \frac{dL(t)}{dt} &= q_A(t) \frac{dq_A(t)}{dt} + \frac{q_B(t)}{y} \frac{dq_B(t)}{dt} = -q_B(t) \\ &+ \begin{cases} q_A(t)(\lambda - 1) & \text{if } r(t)=D \\ q_A(t)(\lambda - x) + q_B(t) \frac{x 1_{[q_A(t) > 0]}}{y} & \text{if } r(t)=B \end{cases}. \end{aligned} \quad (7)$$

Assume $\lambda \geq 1$, else $L(t) < 0$ trivially. With this assumption we have if $r(t) = D$, then

$$\frac{dL(t)}{dt} \leq q_B(t) \frac{x(\lambda-1 - \frac{x-1}{x}y)}{(x-1)y}.$$

Now assume $x \geq y$, else $dL(t)/dt < 0$ trivially. For $r(t) = B$ we have

$$\frac{dL(t)}{dt} \leq q_A(t)(\lambda - (1 + \frac{x-1}{x}y)).$$

Since $\lambda < \min(x, 1 + \frac{x-1}{x}y)$ we have our result.

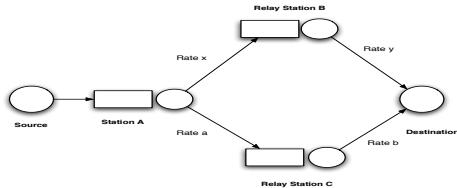


Fig. 3. Two route network.

There is an alternative view of our routing policy that allows us to generalise to the network in Fig. 3. Routing is given by

$$r(t) = \begin{cases} C & \text{if } \frac{q_A(t)}{a} + \frac{q_C(t)}{b} \leq \frac{q_A(t)}{x} + \frac{q_B(t)}{y} \\ B & \text{otherwise} \end{cases} \quad (8)$$

Rearranging terms we get

$$r(t) = \begin{cases} C & \text{if } a(q_A(t) - \frac{xq_C(t)}{b}) \geq x(q_A(t) - \frac{aq_B(t)}{y}) \\ B & \text{otherwise} \end{cases} \quad (9)$$

This resembles a *max-pressure* policy, hence one can show stability using Lyapunov function $L(t) = \frac{q_A^2(t)}{2} + \frac{x}{b} \frac{q_C^2(t)}{2} + \frac{a}{y} \frac{q_B^2(t)}{2}$. Note that since a FIFO two-station tandem network [16] falls under the class we are considering, it is likely finding Lyapunov functions in the general case will be tricky.

V. CONCLUSIONS

We proposed an alternative to *max-pressure* scheduling policies that for interference-free networks resulted in routing packets of the longest queues along (dynamic) shortest path routes. This policy was shown to be stabilising for a class of simple networks. In our simulation experiments this continues to hold for more complex networks.

Thus our future work will be to prove stability for general networks, and to generalise the policy to networks with time-varying rates and interference constraints.

REFERENCES

- [1] L. Georgiadis, M. J. Neely, and L. Tassiulas, “Resource allocation and cross-layer control in wireless networks,” *Foundations and Trends in Networking*, vol. 1, no. 1, pp. 1–149, 2006.
- [2] X. Lin, N. Shroff, and R. Srikant, “A tutorial on cross-layer optimization in wireless networks,” *IEEE JSAC*, pp. 1452–1463, Aug 2006.
- [3] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” *IEEE Transactions on Automatic Control*, vol. 37, pp. 1936–1948, 1992.
- [4] L. Tassiulas and P. Bhattacharya, “Allocation of interdependent resources for maximal throughput,” *Commun. Statist. Stochastic Models*, vol. 16, pp. 27–48, 2000.
- [5] J. G. Dai and W. Lin, “Maximum pressure policies in stochastic processing networks,” *Operations Research*, vol. 53, no. 2, pp. 197–218, 2005.
- [6] A. Stolyar, “Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm,” *Queueing Systems*, vol. 50, pp. 401–457, 2005.
- [7] M. J. Neely, “Optimal backpressure routing for wireless networks with multi-receiver diversity,” in *Proceedings of Conference on Information Sciences and Systems (CISS)*, March 2006.
- [8] T. Ho and H. Viswanathan, “Dynamic algorithms for multicast with intra-session network coding,” in *Proceedings of 43rd Allerton Annual Conference on Communication, Control and Computing*, 2005.
- [9] V. G. Subramanian and D. Leith, “Draining time based scheduling algorithm,” Hamilton Institute, NUIM, in preparation, Tech. Rep., March 2007.
- [10] D. J. Leith and P. Clifford, “Convergence of distributed learning algorithms for optimal wireless channel allocation,” in *Proceedings of IEEE Conference on Decision and Control*, Dec 2006.
- [11] M. R. Garey and D. S. Johnson, *Computers and intractability: A guide to the theory of NP-completeness*. San Francisco, Calif.: W. H. Freeman and Co., 1979.
- [12] J. G. Dai and G. Weiss, “A fluid heuristic for minimizing makespan in job shops,” *Oper. Res.*, vol. 50, no. 4, pp. 692–707, 2002.
- [13] D. Bertsekas and R. Gallager, *Data Networks*, 2nd Ed. Prentice-Hall, 1991.
- [14] H. Chen and D. D. Yao, *Fundamentals of Queueing Networks*. Springer, New York, 2001.
- [15] S. Foss and T. Konstantopoulos, “An overview of some stochastic stability methods,” *Journal of the Operations Research Society of Japan*, vol. 47, no. 4, pp. 275–303, 2004.
- [16] D. Down and S. Meyn, “Piece-wise linear test functions for stability and instability of queueing networks,” *Queueing Systems*, vol. 27, pp. 205–226, 1997.