

Distributed Backpressure Protocols with Limited State Feedback

Scott T. Rager Ertugrul N. Ciftcioglu Aylin Yener Thomas F. La Porta Michael J. Neely*

The Pennsylvania State University, University Park, PA 16802

*University of Southern California, Los Angeles, CA 90089

Email: str5004@psu.edu, enc118@psu.edu, yener@ee.psu.edu, tlp@cse.psu.edu, mjneely@usc.edu

Abstract—The freedom and flexibility of wireless Mobile Ad-hoc Networks (MANETs) that make them extremely desirable for many military, emergency, and sensor network applications also present challenges for multiple layers in the network stack. Max-weight scheduling, also known as backpressure routing, is a cross-layer control algorithm that is well-known to be throughput optimal since it provides queue stability within the network for all traffic injection rates within the network’s capacity. Despite its desirable properties like operating on instantaneous queue and channel states without requiring their statistics, max-weight scheduling relies on global knowledge of full system state information of the network. This is an overly optimistic assumption for most real-world implementations. In this work, we address this issue and develop a distributed max-weight scheduling algorithm in which nodes disseminate the necessary network state information for each to make the same optimal cross-layer control decision individually each time slot. We explore the idea of disseminating only a limited amount of state feedback for this algorithm and evaluate the subsequent impact on performance. We compare the distributed protocol to a centralized version that assumes nodes have global knowledge of this network state information, showing differences in performance from the increased overhead. We also introduce and evaluate an improvement to the algorithm that achieves better performance by dynamically adjusting values used to estimate queue backlogs in the limited state feedback scheduling algorithm.

I. INTRODUCTION

Many military networks today are comprised of wireless nodes that exchange data without the help of external infrastructure like a backbone wired network. These Mobile Ad-hoc Networks (MANETs) provide the desired qualities of robustness and flexibility needed for the unpredictable environments and situations that arise when carrying out tactical missions. The rise in uses for MANETs has garnered a great deal of research aimed at improving their performance. Among the crucial elements affecting this performance are decisions for scheduling the available physical resources and routing packets. The nature of instability of wireless links and thus routes, whether due to fading, interference, or mobility, provides for an argument that the physical layer resource allocation should be jointly considered along with the routing, flow-control, etc. operations of the rest of the network protocol stack.

Research was sponsored by the U.S. Army Research Laboratory under the Network Science Collaborative Technology Alliance, Agreement Number W911NF-09-2-0053.

Accordingly, cross-layer approaches where network control policies make higher-layer routing and resource scheduling choices based on physical layer parameters have been of great interest, e.g., [1]–[3]. For practical networks with stochastic traffic, the *stability region* is the set of arrival rates for which some network control policy that stabilizes all the queues in the network exists. This region provides a fundamental limit for network performance. Any solution to the problem of resource allocation which ensures bounded queues whenever arrival rates lie in the stability region have been termed as *throughput optimal* algorithms [1]–[4]. The essence of these approaches relies on *backpressure routing*, also known as the *Maximum Differential Backlog (MDB)* algorithm [2]. In MDB, the difference in the queue backlogs between transmitting and receiving ends of each link are used as a means of prioritizing the links and performing routing decisions.

Despite its desirable properties, including being throughput-optimal and operating on instantaneous queue and channel states without requiring knowledge of the underlying statistical distributions, max-weight scheduling does require global knowledge of network state information, an unrealistic assumption for most real-world implementations. In practical networks, possessing full system state information requires dissemination of such information to all nodes. Exchanging this network state information leads to increases in protocol complexity and overhead, and, therefore, may result in a reduction in efficiency.

To address this issue, we characterize the effects of disseminating only a limited amount of network state information for use by cross-layer controls algorithms in general multihop networks. In particular, we are interested in the exploration-exploitation trade-off for multihop networks with stochastic traffic. This study’s main focus is the mapping from queue backlog values to their binary queue representations. Using limited amounts of feedback, our goals are to (i) assess the overall effect on delay compared to a network with full queue backlog information and (ii) improve these mappings from queue backlogs to binary representation to improve delay performance.

The remainder of the paper is organized as follows: We discuss related work in Section II. Section III describes the communication model. We present the transmission policies and the protocol details in Section IV. Simulation details are given in V, and numerical results are presented in Section VI.

Finally, we provide the conclusion and future work in Section VII.

II. RELATED WORK

The backpressure algorithm, first introduced in [2], is particularly appealing in the sense that it does not require any a priori information on the input traffic statistics [1], [2], [5]. Extended versions of the MDB have also been used to stabilize networks and minimize a network cost, e.g., average power [6]–[8]. The backpressure policy has also been extended to operate with different physical layer enhancements, e.g., [3], [4], [9]–[13].

The vast field of stochastic network optimization and max weight approaches has been further impacted by [8], which has generalized the applicability of the backpressure approach to networks with arbitrary sample paths for arrivals, channels and mobility. The algorithm proposed provides network-throughput utility that is arbitrarily close to that of a T-slot lookahead policy, with a trade-off in the worst-case backlog kept in any queue.

In [1], it is shown that even if inexact or estimated values of queue backlogs are used, the backpressure algorithm achieves full throughput with a tradeoff in the average delay. In particular, the average delay increases proportionally with the estimation error as long as the difference between the estimates of the queue information and the actual queue values is bounded.

There have been attempts to simplify the centralized scheduling with full knowledge. The majority of these approaches follow approximation of the backpressure algorithm with maximal and greedy scheduling [14], [15]. There have also been studies which concentrate on limited channel state information [16], like receiving feedback from a subset of users [17], for example.

In [7], it is observed that only one bit of queue information of the other source leads to significant improvement compared with no information for energy minimization in a bidirectional relay network where the relay applies network coding. This approach can be classified as reducing the bit-complexity [18] of message passing. In addition to simplifying the process of feedback, it can reduce the resources and time allocated for control signalling.

We note that the tradeoff between improved scheduling due to extra information and the overhead required to deliver it has recently [19] been considered, but for a significantly different model. In particular, an infinitely backlogged system model is considered with no stochastic arrivals. The improvement on scheduling is based on a collision avoidance basis, which is considerably different than our goal of approximating the maximum possible weighted rate sum.

III. SYSTEM MODEL

While the concepts of limited state feedback and its use in max-weight scheduling can be applied to any network topology, the specific amount of overhead incurred depends on the specific protocol used for disseminating this control

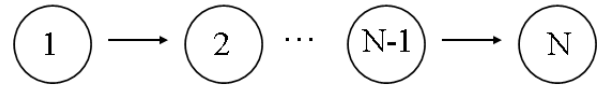


Fig. 1. N-node line network.

information. As an illustrative network topology that necessitates a significant amount of control packet exchanges for full dissemination of network state information, in this work we consider the fundamental network of a multi-hop line network. In this line network, Node 1 is the only source node, and Nodes 2,3,...,N−1 forward traffic from 1 to its destination at N. Since no direct link exists between these end nodes, the relays enable communication. Decode-and-forward relaying is used [20]. We assume that nodes are half-duplex, i.e., they cannot transmit and receive at the same time. Moreover, we assume that interference is negligible for a distance of two hops.

Accordingly, two activation sets are defined: S_1 with nodes $\{1, 3, 5, \dots\}$ active, and S_2 with nodes $\{2, 4, 6, \dots\}$ active. The network operates in time slots. At the beginning of each slot, decisions are made regarding which nodes will transmit and with what rates. For this scenario, the main issue is scheduling, i.e., determining which of the two activation sets will be activated at each time slot t .

Traffic arrives at node i according to an ergodic process with average rate λ_i packets/second. The arrival rates in terms of packets/second can be converted to arrival rates ρ_i in bits/second via $\rho_i = \lambda_i E[L_i]$, where L_i denotes lengths of packets arriving at node i .

The queue dynamics of node i can be expressed as:

$$q_i(t+1) = \max(q_i(t) - r_i(t), 0) + a_i(t), \quad (1)$$

where $q_i(t)$ is the queue size at node i in bits at the beginning of slot t , $r_i(t)$ is the total data rate leaving node i in slot t , and $a_i(t)$ is the number of bits arrivals in slot t . We note that $a_i(t)$ includes both exogenous arrivals to the network and endogenous arrivals from other users in the network for routing.

IV. SCHEDULING WITH LIMITED INFORMATION

Consider a general multiuser topology. Depending on the physical layer techniques and properties, the set of simultaneously activated link rates will be given by the feasible rate region. Let us focus on a network with slotted operation, and let $\mathcal{R}(t)$ denote the set of all feasible link rates $\{r_{ij}(i,j) \in \mathcal{L}\}$ in slot t . For the line network considered here, $\mathcal{R}(t)$ is $(N-1)$ -dimensional, consisting of the union of $\{r_{12}(t), 0, r_{34}(t), 0, \dots\}$ and $\{0, r_{23}(t), 0, r_{45}(t), \dots\}$, where $r_{ij}(t)$ is the maximum supported rate of link (i,j) at slot t .

With full state information, the throughput optimal maximum weight policy performs rate allocation and scheduling in order to:

$$\max_{r_{ij} \in \mathcal{R}(t)} \sum_{(i,j) \in \mathcal{L}} (q_i(t) - q_j(t)) r_{ij}(t), \quad (2)$$

where (i,j) denotes the link from node i to j and \mathcal{L} is the set of all links.

Let $W^*(t)$ be the maximum sum achieved. Note that $W^*(t)$ is achieved with the proper rates $r_{i,j}^*(t) \in \mathcal{R}(t)$ for each link (i, j) . On the other hand, users have estimates of the queues, $\hat{q}_k(t)$, instead of $q_k(t)$ as a result of only exchanging limited information.

Applying the backpressure algorithm using these estimates, users would solve

$$\max_{r_{i,j} \in \mathcal{R}(t)} \sum_{(i,j) \in \mathcal{L}} (\hat{q}_i(t) - \hat{q}_j(t)) r_{ij}(t), \quad (3)$$

possibly resulting in a different set of activated nodes with rates $\tilde{r}_{i,j}(t) \in \mathcal{R}(t)$.

The actual weighted rate sum that would result from rate allocation $\tilde{r}_{i,j}(t)$ would be

$$W(t) \triangleq \sum_{(i,j) \in \mathcal{L}} (q_i(t) - q_j(t)) \tilde{r}_{ij}(t), \quad (4)$$

In [18] it is shown that as long as

$$W(t) \geq G(t)W^*(t) - C(t) \quad (5)$$

where $G(t)$ and $C(t)$ are random variables such that

$$G(t) \geq \gamma, \text{ and } E[C(t)|\mathbf{Q}(t)] \leq \xi \quad (6)$$

where $\mathbf{Q}(t)$ is the vector of queue states at each node, $W^*(t)$ is the maximum weight (weighted sum rates) with full queue and channel information, and $W(t)$ is the maximum weight with partial information, the throughput depends on γ and, the average delay upper bounds depend on ξ .

From (4) and (2) we have $W(t)$ and $W^*(t)$ applicable to the form of (5). Moreover, from [1] it is known that as long as the error in queue estimates is bounded for each slot t , it is possible to achieve the full throughput region ($\gamma = 1$), however the upper bound on average delay increases proportionally with ξ .

As a result, minimizing the degradation in average delay involves the following objective:

$$\min_{\tilde{r}_{ij}(t) \in \mathcal{R}(t)} \sum_{(i,j) \in \mathcal{L}} (q_i(t) - q_j(t))(r_{ij}^*(t) - \tilde{r}_{ij}(t)), \quad (7)$$

where $r_{ij}^*(t)$ and $\tilde{r}_{ij}(t)$ are defined from (3)-(4). It is obvious that $\tilde{r}_{ij}(t)$ depends on the queue estimates $\hat{q}(t)$, which in turn depend on the mapping defined for conveying the limited queue information. For this reason, we focus on how queue estimates rely on the amount of control information exchanged. This demonstrates the impact on throughput due to overhead and the impact on average delay through both overhead and imperfect scheduling due to limited state feedback.

A. Protocol Details

The amount of control packet payload needed to disseminate global queue backlog greatly varies on the topology. In our motivating example of a multihop line network of N users, data exchange occurs in two phases. First, packets are broadcast by successive nodes starting with node 1 and ending with node $N - 1$, with each node appending its own

queue information. Then the process is repeated in reverse order. Each control packet is decoded by the two neighbors of the transmitting node. This leads to the following amount of required queue exchanges:

$$1 + 2 + \dots + (N - 2) + 1 + 1 + 2 + \dots + (N - 3) \quad (8)$$

$$= \frac{(N - 2)(N - 1)}{2} + 1 + \frac{(N - 3)(N - 2)}{2} \quad (9)$$

$$= (N - 2) \frac{2N - 4}{2} + 1 = (N - 2)^2 + 1, \quad (10)$$

meaning a feedback requirement almost on the order of N^2 .

Accordingly, queue information feedback can lead to significant inefficiency. On the other hand, for clustered networks like an uplink system with simultaneous transmission, the amount of queue info conveyed is in the order of number of users. Hence, finer quantization levels can be afforded. Each queue calculates the rate allocation solution by (4) with the amount of limited feedback disseminated by the network.

The method that we consider for queue information estimates is to utilize feedbacks of K bits in order to represent the queue state. More specifically, using K bits, we can specify where the queue backlog value falls within $2^K - 1$ different threshold values.

Since there is no problem of rate allocation for scheduled nodes for the line network described in Section III, the max-weight scheduling algorithm with limited queue information results in incorrect decisions whenever the wrong activation set is scheduled because $\hat{q}_i - \hat{q}_j$ differs from $q_i - q_j$.

Considering the case for $K = 2$ provides insight into the tradeoff between accurate information and amount of feedback. With 2 bits, node k can refine the granularity of its queue information to one of four possible values, $\{Q_1, Q_2, Q_3, Q_4\}$ specifying where it falls between three threshold values, $\{T_1, T_2, T_3\}$. In this case, $\hat{q}_i - \hat{q}_j$ can now take one of seven values. In general, K bits provides a level of granularity of 2^K for reporting queue levels, resulting in $2^{K+1} - 1$ possible differential values.

B. Dynamic Threshold Adaptation

We propose a threshold updating mechanism where the queue feedbacks at any slot are used to adapt the quantization levels for the next time slot. In a view to better differentiate among queues with the given quantization levels, we lower the step size whenever the highest possible quantization level is not utilized.

Let U denote the maximum range of queue size for quantization at any slot. We distribute the $2^K - 1$ threshold values uniformly within U , such that each quantization step has a size of $\frac{U}{2^K}$, and the queue value estimate can be calculated by $\hat{q}_i = \lfloor q_i / (\frac{U}{2^K}) \rfloor$. The specific description of our threshold adaptation algorithm is given as follows:

- 1) Initialize $U = \text{Max Buffer Size}$
- 2) At each time slot t , $t = 2, \dots$, observe all conveyed backlogs \hat{q}_i , $i = 1, 2, \dots, N$
if $\forall i, \hat{q}_i \neq 2^K - 1$
decrease U to $\max(2^K, U - \frac{U}{2^K})$

else,

increase U to $\min(\text{Max Buffer Size}, U + \frac{U}{2^K})$

The intuition behind the algorithm is as follows: If the uppermost quantization level is not used, we expect that a finer quantization can be performed by reducing the quantization granularity. Note that any reduction in step sizes is lower bounded by 1. On the other hand, if any queue conveys the maximum possible feedback state, we increase the quantization step sizes due to the possibility that the range provided is not sufficient. Note also that the maximum quantization level is bounded by the buffer size.

V. SIMULATION DETAILS

A max-weight scheduling algorithm designed within the framework of [8] was implemented in the Qualnet¹ network simulation environment to evaluate the effect of disseminating network state information as well as the possible gains achieved through utilizing limited state feedback.

The simulation scenario mimics the network model described in Section III. A time-slotted system is used with nodes utilizing a fraction of time, T_C , at the beginning of each slot to exchange necessary *control* information, e.g., queue backlogs. The remainder of the time slot, T_D , is then utilized for *data* packet transmission by the nodes in the activation set chosen by the algorithm in that time slot. T_D is constant in all trials, while T_C , and, therefore, the overall time slot duration, $T_{slot} = T_C + T_D$, depends on the amount of control information disseminated, as in (8). Results are reported in units of packets per second to fairly compare different runs.

The scenario simulated is a 5 node line network with the same assumptions described above in Section III. An application at node 1 generates traffic of 32 byte packets at an average rate of λ packets per second. When these packets arrive at the destination node, 5, they are removed from the network immediately. Here, for simplicity, queue backlogs and rates are treated using units of packets and packets per time slot rather than bit and bits per second.

We assume that each link from node n to $n + 1$ achieves a viable rate of $\{2, 4, 6\}$ packets with respective probabilities of $\{0.3, 0.3, 0.4\}$. We also assume that these channel states only change between time slots, not during, so that nodes can transmit the number of packets according to its channel rate during the time slot without loss. Data rates for the exchange of control packets are assumed to be a constant 10 times slower than the highest achievable data transmission rate to ensure reliable transmission. T_C is calculated for each run, then, based on this data rate and the amount of data being disseminated for that run.

VI. NUMERICAL RESULTS

A. Varying Number of Bits for Queue Representation

The first set of experiments examine the tradeoff between the overhead of exchanging network state information and the gains in reducing packet delay from better scheduling

decisions in the max-weight algorithm due to more accurate queue representations. Queue backlogs are represented by K bits. The values are calculated using threshold values uniformly distributed throughout the size of buffers in the network. Here, we assume that buffers are infinitely large so no packets are lost due to overflow. For calculation of queue estimates, however, we use a maximum buffer size of 8192 packets, meaning a value of $K = 13$ bits results in disseminating the exact queue backlogs. For $K = 1$, in this case, nodes only report if their backlogs are above or below 4096. Using this scheme, the maximum error in queue estimates is bounded by 2^{13-K} .

We simulated the line network described above using an arrival rate of 1 packet every 4.3 milliseconds. We set $T_D = 10$ milliseconds and assume a maximum data transmission rate of 1 Mbps. Each run simulated 1,000 seconds of network time (approximately 100,000 time slots) with statistics recorded and averaged over the last 25% of the run to capture the steady state conditions of the network.

We varied the number of bits used to represent queue backlogs, K , from 1 to 13. For comparison, we also show results assuming global knowledge of exact queue backlogs without the cost of exchange overhead, i.e., $T_C = 0$.

Figure 2 shows the throughput achieved for each value of K in packets per second as well as the throughput achieved assuming global knowledge. Here, as we expect, throughput is essentially optimal for low amounts of feedback since the error in queue backlog representation is bounded. As the level of feedback increases, however, we witness the impact of increasing overhead for control information dissemination and its effect on network capacity. For $K \leq 9$, the network is able to operate at its full capacity, delivering a consistent throughput even for the case where the network exchanges only $K = 1$ bit for each queue backlog. For these values of K , there is enough capacity remaining in each time slot for data transmission so that the system is stable in terms of queue length. For values of $K \geq 10$, however, T_C grows prohibitively large, shrinking the network's effective capacity and thus the achievable throughput. In these cases, data is being buffered in the network and the queue occupancy starts to grow because there is not enough capacity in the network to carry the offered load.

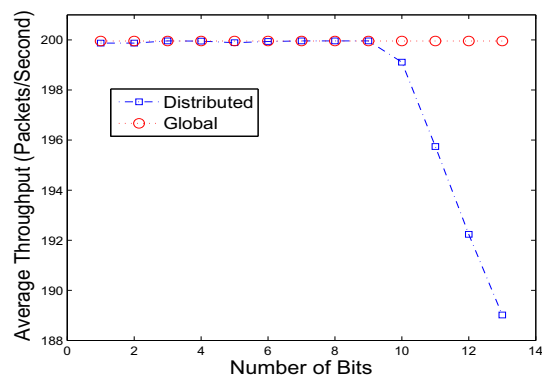


Fig. 2. Throughput achieved for different levels of state feedback.

¹<http://www.scalable-networks.com>

Figure 3 shows the average total backlog of queues in the network, a value which is of interest because it is proportional to packet delay according to Little's Law [1]. Here, we clearly see the tradeoff in better scheduling decisions and the amount of overhead required for varying levels of feedback. The delay decreases considerably until reaching a saturation point of $K = 9$. After this point, as we observed with throughput, the proportion of time consumed by exchanging control information becomes prohibitive, causing the network capacity to drop and queue sizes to grow.

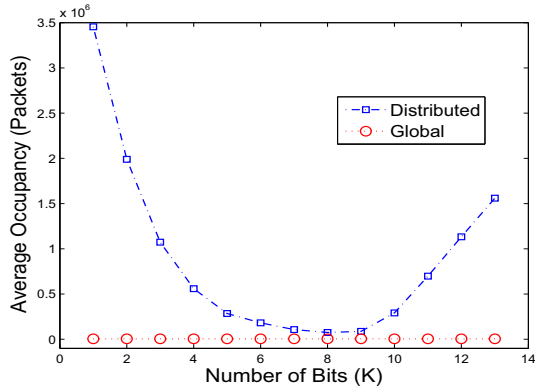


Fig. 3. Average total backlog in network for different levels of state feedback.

B. Applying Adaptive Threshold Algorithm

Next, we present results for the same scenario using the adaptive threshold algorithm presented in Section IV-B. Figure 4 shows a comparison of the max-weight scheduling algorithm using the standard limited feedback scheme and then using adaptive thresholds. Here, we can see that adaptive thresholds decrease the resulting delay, actually reaching its minimum value at $K = 3$. At this value, the gain from exchanging less control information allows the network to forward traffic efficiently enough to establish a backlog gradient using an average value of just 14.1 for the maximum buffer size, U .

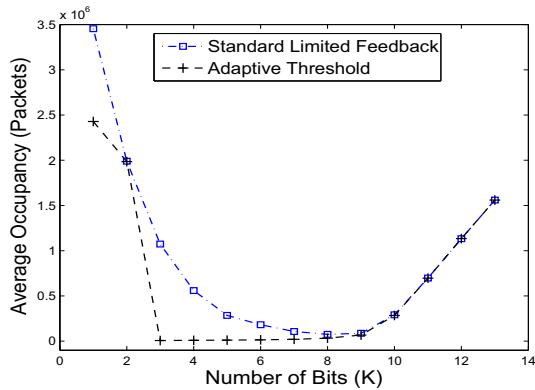


Fig. 4. Comparison of average total backlog using standard and adaptive thresholds.

C. Performance of Realistic Time Scale

The results above show the performance of the algorithm after its steady state operating point has been reached. While these results show the importance and potential gains for both limited state feedback and adaptive thresholds, the scenario is

not a realistic representation of one that would be found in a realistic military network.

To gain better understanding of the benefits and costs of exchanging more accurate queue backlogs, we collected statistics of the same network scenario over just 100 seconds. This case is representative of a network that sustains short-lived flows. Here, the results are also collected over the entire simulation time to show the impact of time needed to build the backpressure gradient to achieve the steady state throughput.

Figure 5 shows the throughput simulating the standard limited feedback algorithm for varying values of K . For $K = 1$, no differential backlog is seen by the nodes until nodes reach a value of 4096. Even though a small fraction of each time slot is devoted to control overhead, the time spent on establishing the backlog gradient during which data is not consistently delivered to the destination, drastically reduces the average throughput received over the simulation duration. Conversely, for $K = 13$, the backpressure provides a route much more quickly, but the extra overhead reduces effective throughput.

The related delay for this scenarios is shown is figure 6. Naturally, the same trend in performance is seen, with delay reaching a minimum at $K = 9$.

Figures 5 and 6 also show throughput and average total occupancy, respectively, simulating the same network with adaptive thresholds implemented. We can see that introducing the adaptive threshold leads to gains for all levels of feedback, especially $K \geq 3$. These results show the potential for max-weight scheduling algorithms to be effective in actual applications.

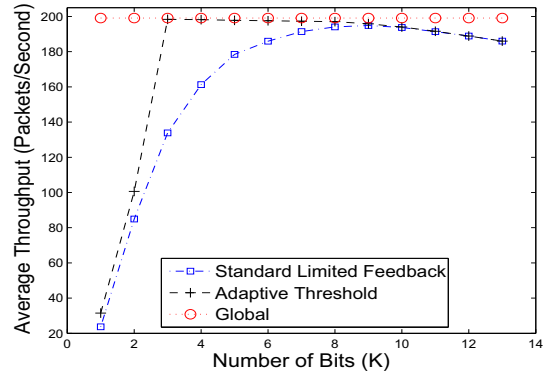


Fig. 5. Throughput for simulation with realistic time-frame.

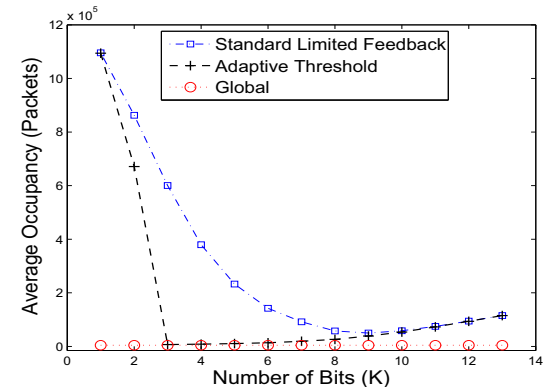


Fig. 6. Average total backlog for simulation with realistic time-frame.

D. Comparison of Different Maximum Buffer Sizes

Intuitively, a larger maximum buffer size will require larger queue backlogs to build up before establishing a gradient along which packets will be routed when using limited state feedback with static threshold values. In this case, performance achieved over a short period of time will suffer from waiting for this backpressure to build up. Figure 7 provides evidence to this effect as the average total number of packets in the network for a 16384 packet buffer is consistently larger than that of the same scenario using a 8192 packet buffer.

Figure 7 also shows the benefit of the adaptive threshold algorithm. Using this scheme, average delay for both buffer sizes are almost identical. This result is also intuitive; using adaptive thresholds allows the limited state feedback to set thresholds based on the effective queue backlogs in the network since unused portions of the buffers are not considered when calculating estimates.

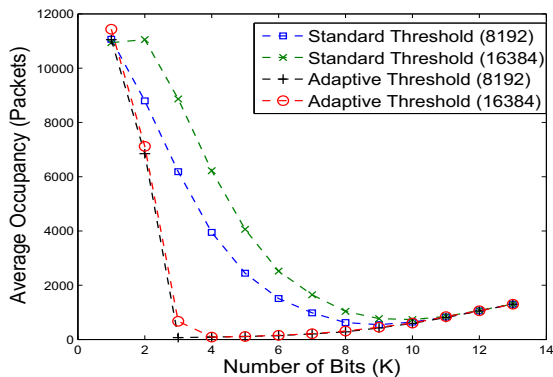


Fig. 7. Average total backlog for different buffer sizes using both static and adaptive thresholds.

VII. CONCLUSION

In this paper, we have studied the use of max-weight scheduling for making decisions across several layers of the network protocol stack. Although proven to be throughput optimal, we have demonstrated that this approach has several drawbacks, including large potential delays, the need for global network state information, and long convergence times. To approach realistically implementable backpressure routing algorithms with more desirable performance characteristics, we have presented a modified version of the max-weight optimization problem that uses limited network state feedback by dividing queues using chosen threshold values.

Using the Qualnet network simulation environment, we have simulated this limited state feedback approach. From these simulations, we observe that when using uniformly distributed threshold values, a tradeoff exists between the overhead required in information dissemination and performance gains from more effective scheduling. We also present an adaptive threshold algorithm that reduces delay by dynamically adjusting these threshold levels. Finally, we provide results for a simulation of a network in a realistic time-scale for military networks and show the performance gains that can be achieved by using the limited state feedback with adaptive threshold approach.

In this work, we considered a line network topology. Future work involves investigation and performance evaluation of distributed backpressure with limited state feedback for arbitrary topologies and mobile nodes.

REFERENCES

- [1] L. Georgiadis, M. J. Neely, and L. Tassiulas. Resource Allocation and Cross-layer Control in Wireless Networks. *Foundations and Trends in Networking*, 2006.
- [2] L. Tassiulas and A. Ephremides. Stability Properties of Constrained Queuing Systems and Scheduling Properties for Maximum Throughput in Multihop Radio Networks. *IEEE Transactions on Automatic Control*, 37(12):1936–1948, December 1992.
- [3] E. M. Yeh and R. A. Berry. Throughput Optimal Control of Cooperative Relay Networks. *IEEE Transactions on Information Theory: Special Issue on Models, Theory, and Codes for Relaying and Cooperation in Communication Networks*, 53(10):3827–3832, October 2007.
- [4] T. J. Oechtering and H. Boche. Stability Region of an Efficient Bidirectional Regenerative Half-duplex Relaying Protocol. In *Proc. IEEE Information Theory Workshop, 2006. ITW '06*, Chengdu, China, October 2006.
- [5] M. J. Neely, E. Modiano, and C.E. Rohrs. Dynamic Power Allocation and Routing for Time Varying Wireless Networks. *IEEE Journal on Selected Areas in Communications, Special Issue on Wireless Ad-Hoc Networks*, 23(1):89–103, March 2005.
- [6] M. J. Neely. Energy Optimal Control for Time Varying Wireless Networks. *IEEE Transactions on Information Theory*, 52(7):2915–2934, July 2006.
- [7] E. N. Ciftcioglu, Y. E. Sagduyu, R. A. Berry, and A. Yener. Cost Sharing with Network Coding for Two-Way Relay Networks. In *Proc. Forty-Seventh Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, September 2009.
- [8] M. J. Neely. Universal scheduling for networks with arbitrary traffic, channels, and mobility. *Proc. IEEE Conf. on Decision and Control (CDC)*, Atlanta, GA, Dec. 2010.
- [9] M. J. Neely and R. Uргаonkar. Optimal Backpressure Routing for Wireless Networks with Multi-Receiver Diversity. *Ad Hoc Networks (Elsevier)*, 7(5):862–881, March 2009.
- [10] Y. E. Sagduyu, D. Guo, and R. A. Berry. Throughput Optimal Control for Relay-Assisted Wireless Broadcast with Network Coding. In *Proc. IEEE International Workshop on Wireless Network Coding*, San Francisco, CA, June 2008.
- [11] M. J. Neely and A. Sharma. Dynamic Data Compression with Distortion Constraints for Wireless Transmission over a Fading Channel. *arXiv:0807.3768v1*, submitted, July 2008.
- [12] E. N. Ciftcioglu, A. Yener, and R. A. Berry. Stability Regions of Bidirectional Cooperative Relaying. *Submitted to IEEE Transactions on Information Theory*, September 2010.
- [13] E. N. Ciftcioglu, Y. E. Sagduyu, A. Yener, and R. A. Berry. Queue-based Compression in a Two-Way Relay Network. In *Proc. Conference on Information Sciences and Systems*, Princeton, NJ, March 2010.
- [14] P. Chaporkar, K. Kar, and S. Sarkar. Throughput guarantees through maximal scheduling in wireless networks. In *In Proceedings of 43rd Annual Allerton Conference on Communication, Control and Computing*, pages 28–30, 2005.
- [15] X. Lin and N.B. Shroff. The impact of imperfect scheduling on cross-layer rate control in wireless networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 3, pages 1804 – 1814 vol. 3, march 2005.
- [16] M. Ouyang and L. Ying. On Scheduling in Multi-channel Wireless Downlink Networks with Limited Feedback. In *Proc. Allerton*, 2009.
- [17] A. Gopalan, C. Caramanis, and S. Shakkottai. On Wireless Scheduling for Partial Channel state Information. In *Proc. Allerton Conference on Communication, Control and Computing*, September 2007.
- [18] Y. Yi, A. Poutiere, and M. Chiang. Complexity of Wireless Scheduling: Impacts and Tradeoffs. *ACM MobiHoc*, May 2008.
- [19] J. Hong and V. O. K. Li. The Effect of Information on Scheduling Performance in Multihop Wireless Networks. *IEEE Transactions on Wireless Communications*, 9(10), October 2010.
- [20] T. M. Cover and A. El Gamal. Capacity Theorems for the Relay Channel. *IEEE Transactions on Information Theory*, 25(5):572–584, September 1979.