

Delay Analysis and Optimality of Scheduling Policies for Multi-Hop Wireless Networks

Gagan Raj Gupta, Ness B. Shroff

Abstract—In this paper, we analyze the delay performance of a multi-hop wireless network in which the routes between source-destination pairs are fixed. We develop a new queue grouping technique to handle the complex correlations of the service process resulting from the multi-hop nature of the flows and their mutual sharing of the wireless medium. A general set-based interference model is assumed that imposes constraints on links that can be served simultaneously at any given time. These interference constraints are used to obtain a fundamental lower bound on the delay performance of any scheduling policy for the system. We present a systematic methodology to derive such lower bounds. For a special wireless system, namely the clique, we design a policy that is sample path delay optimal. For the tandem queue network, where the delay optimal policy is known, the expected delay of the optimal policy numerically coincides with the lower bound. The lower bound analysis provides useful insights into the design and analysis of optimal or nearly optimal scheduling policies. We conduct extensive numerical studies to demonstrate that one can design policies whose average delay performance is close to the lower bound computed by the techniques presented in this paper.

I. INTRODUCTION

A large number of studies on multi-hop wireless networks have been devoted to system stability while maximizing metrics like throughput or utility. These metrics measure the performance of a system over a long time-scale. For a large class of applications such as video or voice over IP, embedded network control and for system design; metrics like delay are of prime importance. The delay performance of wireless networks, however, has largely been an open problem. This problem is notoriously difficult even in the context of wireline networks, primarily because of the complex interactions in the network (e.g., superposition, routing, departure, etc.) that make its analysis amenable only in very special cases like the product form networks. The problem is further exacerbated by the mutual interference inherent in wireless networks which, complicates both the scheduling mechanisms and their analysis. Some novel analytical techniques to compute useful lower bound and delay estimates for wireless networks with single-hop traffic were developed in [12]. However, the analysis is not directly applicable to multi-hop wireless network with multi-hop flows, due to the difficulty in characterizing the departure process at intermediate links. The metric of interest in this paper is the system-wide average delay of a packet from the

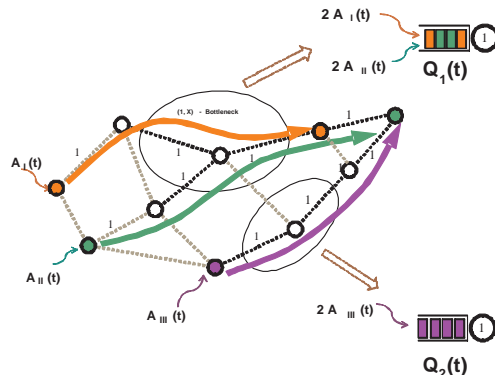


Fig. 1. A typical multi-hop wireless network with multiple flows, each having exogenous arrivals at the source. Some of the important bottlenecks have been highlighted.

source to its corresponding destination. We present a new, systematic methodology to obtain a fundamental lower bound on the average packet delay in the system under any scheduling policy. Furthermore, we re-engineer well known scheduling policies to achieve good delay performance viz-a-viz the lower bound.

In this paper, we analyze a multi-hop wireless network with multiple source-destination pairs, given routing and traffic information. Each source injects packets in the network, which traverses through the network until it reaches the destination. For example, a multi-hop wireless network with three flows is shown in Fig. 1. The exogenous arrival processes $A_I(t)$, $A_{II}(t)$ and $A_{III}(t)$ correspond to the number of packets injected in the system at time t . A packet is queued at each node in its path where it waits for an opportunity to be transmitted. Since the transmission medium is shared, concurrent transmissions can interfere with each others' transmissions. The set of links that do not cause interference with each other can be scheduled simultaneously, and we call them *activation vectors* (matchings). We do not impose any a priori restriction on the set of allowed activation vectors, i.e., they can characterize any combinatorial interference model. For example, in a K -hop interference model, the links scheduled simultaneously are separated by at least K hops. In the example show in Fig. 1, each link has unit capacity; i.e., at most one packet can be transmitted in a slot. For the above example, we assume a 1-hop interference model.

The delay performance of any scheduling policy is primarily limited by the interference, which causes many bottlenecks to be formed in the network. We demonstrated the use of exclusive sets for the purpose of deriving lower bounds on delay for a wireless network with single hop traffic in [12]. In this paper, we further generalize the typical notion of a

Gagan Raj Gupta is with the Department of Computer Science, University of Illinois at Urbana Champaign, Urbana IL 61801 USA e-mail: gagan@illinois.edu

Ness B. Shroff is with the Departments of ECE and CSE, The Ohio State University, Columbus, Ohio, USA e-mail: shroff@ecn.osu.edu

This work was supported in part by ARO W911NF-08-1-0238 and NSF award 07221236-CNS. A preliminary version of this paper appeared in INFOCOM 2009.

bottleneck. In our terminology, we define a (K, X) -bottleneck to be a set of links X such that no more than K of them can simultaneously transmit. Figure 1 shows $(1, X)$ bottlenecks for a network under the 1-hop interference model. In this paper, we develop new analytical techniques that focus on the queueing due to the (K, X) -bottlenecks. One of the techniques, which we call the “*reduction technique*”, simplifies the analysis of the queueing upstream of a (K, X) -bottleneck to the study of a single queue system with K servers as indicated in the figure. Furthermore, our analysis needs only the exogenous inputs to the system and thereby avoids the need to characterize departure processes on intermediate links in the network. For a large class of input traffic, the lower bound on the expected delay can be computed using only the statistics of the exogenous arrival processes and not their sample paths. To obtain a lower bound on the system wide average queueing delay, we analyze queueing in multiple bottlenecks by relaxing the interference constraints in the system. Our relaxation approach is novel and leads to non-trivial lower bounds.

It is also possible to derive stochastic upper bounds on the average delay of the network using the techniques of Lyapunov drifts [8]. We were able to obtain sharper upper bounds in [12] by using a different Lyapunov function, but we do not pursue them here because they focus on a specific scheduling scheme. Our focus on the other hand, is to derive a fundamental bound on the performance of any policy. Moreover, our lower bound techniques captures the effect of statistical multiplexing of packets due to several flows passing through a common (K, X) -bottleneck, which cannot be analyzed using the method of Lyapunov drifts. As a result, the upper bounds computed using these techniques [8] tend to be quite loose in most practical scenarios [12].

We consider the lower bound analysis as an important first step towards a complete delay analysis of multi-hop wireless systems. For a network with node exclusive interference, our lower bound is tight in the sense that it goes to infinity whenever the delay of any throughput optimal policy is unbounded. For a tandem queueing network, the average delay of a *delay optimal* policy proposed by [31] numerically coincides with the lower bound provided in this paper. A clique network is a special graph where at most one link can be scheduled at any given time. Using existing results on work conserving queues, we design a delay optimal policy for a clique network and compare it to the lower bound.

We will see that although delay optimal policies can be derived for some simple networks like the clique and the tandem, deriving such policies in general, is extremely complex. Instead, we re-engineer a well known throughput optimal scheduling policy known as the back-pressure policy and demonstrate that for certain representative topologies, its delay performance is close to the fundamental lower bound. Finally, we also present a case where neither back-pressure policy nor the shadow queue approach (proposed in [2]) are close to the lower bound. For this case, we design a new hand-crafted policy whose delay performance is actually close to the lower bound, thus demonstrating that the lower bound analysis provides useful insights into the design and analysis of optimal

or nearly optimal scheduling policies.

We now summarize our main contributions in this paper:

- Development of a new queue grouping technique to handle the complex correlations of the service process resulting from the multi-hop nature of the flows. We also introduce a novel concept of (K, X) -bottlenecks in the network.
- Development of a new technique to reduce the analysis of queueing *upstream* of a bottleneck to studying simple single queue systems. We derive sample path bounds on a group of queues upstream of a bottleneck.
- Derivation of a fundamental lower bound on the system-wide average queuing delay of a packet in multi-hop wireless network, regardless of the scheduling policy used, by analyzing the single queue systems obtained above.
- Extensive numerical studies and discussion on useful insights into the design of optimal or nearly optimal scheduling policies gained by the lower bound analysis.

We begin with the description of the system model. We then present our methodology for obtaining reductions and using them to lower bound the system-wide average delay of packets. We next address the question of designing delay-efficient schedulers. We then provide concrete examples illustrating the methodology and comparison of the back-pressure policy to the lower bound. We also describe how the proposed approach differs fundamentally from the existing techniques and can be used to gain deeper understanding of the scheduling policies for wireless networks.

II. SYSTEM MODEL

We consider a wireless network $G = (V, L)$, where V is the set of nodes and L is the set of links. Each link has unit capacity. There are N flows, each distinguished by its source destination pair (s_i, d_i) . There is a fixed route (set of links) between the source s_i and corresponding destination d_i . Each route is a simple path. Each flow has its own exogenous arrival stream $\{A_i(t)\}_{t=1}^{\infty}$.

Each packet has a deterministic service time equal to one unit. The exogenous arrivals at each source are assumed to be independent. Let $\mathbf{A}(t) = (A_1(t), \dots, A_N(t))$ represent the vector of exogenous arrivals, where $A_i(t)$ is the number of packets injected into the system by the source s_i during time slot t (for $i \in 1, \dots, N$). Let $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_N)$ represent the corresponding long-term average arrival rate vector.

The path on which flow i is routed is specified as $P_i := (v_i^0, v_i^1, \dots, v_i^j, \dots, v_i^{|P_i|})$, where v_i^j is a node at a j -hop distance from the source node s_i . The source node s_i is denoted by v_i^0 and the destination node d_i by $v_i^{|P_i|}$, where $|P_i|$ is the path length. The packets arriving at each node are queued. Each node maintains a separate queue for each flow that passes through the node. Let $Q_i^j(t)$ denote the queue length at node v_i^j corresponding to flow i . After reaching the destination node, each packet leaves the system, i.e. $Q_i^{|P_i|} = 0$. The queue length vector is denoted by $\mathbf{Q}(t) = (Q_i^j(t) : i \in \{1, 2, \dots, N\} \text{ and } j \in \{1, 2, \dots, |P_i|\})$. Multiple flows can share a link e . A link can be activated in a time slot t only if the corresponding

queue is non empty. We use the term activation (scheduling) of a link or a queue interchangeably. At most, one packet is served at a queue in a given time slot. The service structure is slotted.

The set of links that do not cause mutual interference and hence can be scheduled simultaneously are called activation vectors (matchings). Let \mathcal{J} be the collection of all activation vectors \mathbf{J} . We allow the activation vectors to be arbitrary i.e., they can characterize any interference model. At each time-slot an activation vector $\mathbf{I}(t)$ is scheduled depending on the scheduling policy and the underlying interference model. The indicator function $I_i^j(t)$ indicates whether or not flow i received service at the $(j+1)^{\text{st}}$ hop from source s_i at time slot t . Note that

$$I_i^j(t) = \begin{cases} 1 & \text{if } Q_i^j(t) > 0, \text{ and link } e = (v_i^j, v_i^{j+1}) \text{ is scheduled} \\ 0 & \text{otherwise} \end{cases} \quad (\text{II.1})$$

The evolution of the queues in the system is as follows,

$$Q_i^j(t+1) = \begin{cases} Q_i^j(t) - I_i^j(t) + I_i^{j-1}(t) & \text{if } j > 0 \\ Q_i^j(t) - I_i^j(t) + A_i(t) & \text{otherwise} \end{cases} \quad (\text{II.2})$$

We use the 2-hop interference model in most of our simulation studies since it has often been used to model the behavior of a large class of MAC protocols based on virtual carrier sensing using RTS/CTS messages, which includes the IEEE 802.11 protocol [1]. Under an h -hop interference model, any two active links in $\mathbf{I}(t)$ are always separated by h or more hops in the underlying network graph.

III. DERIVING LOWER BOUNDS ON AVERAGE DELAY

In this section, we present our methodology to derive lower bounds on the system-wide average packet delay for a given multi-hop wireless network. At a high level, we partition the flows into several groups. Each group passes through a (K, X) -bottleneck and the queueing for each group is analyzed individually. The grouping is done so as to maximize the lower bound on the system wide expected delay. For flows passing through a given bottleneck (a group), we lower bound the sum of queues upstream and downstream of the bottleneck separately. We *reduce* the analysis of queueing *upstream* of a (K, X) -bottleneck to studying single queue systems fed by appropriate arrival processes. These arrival processes are simple functions of the exogenous arrival processes of the original network. For example, Figure 1 shows the reduction of two $(1, X)$ bottlenecks in the network. A separate lower bound can be established for the queues downstream of the network. The lower bound on the system-wide average delay of a packet is then computed using the statistics of the exogenous arrival processes. We derive analytical expressions of the lower bounds for a large class of arrival processes.

In this section, we first characterize the bottlenecks in the system. We then explain how to lower bound the average delay of the packets of the flows that pass through a given (K, X) -bottleneck. Our analysis justifies the reduction of a (K, X) -bottleneck to a single queue system fed by appropriate arrival processes. Finally, we present a greedy algorithm which takes as input, a system with N flows and possibly multiple

bottlenecks, and returns a lower bound on the system-wide average packet delay.

A. Characterizing Bottlenecks in the system

Link interference causes certain bottlenecks to be formed in the system. We define a (K, X) -bottleneck to be a set of links $X \subset L$ such that no more than K of its links can be scheduled simultaneously. For example, [12], [15] identify cliques in the conflict graph as the bottlenecks. This corresponds to a set of links, among which only one link can be scheduled at any given time. We call these sets of links *exclusive sets*. We also discuss another type of bottleneck in the case of a cycle graph with 5 nodes, where no more than two links can be scheduled simultaneously. Some of the important exclusive sets for the wireless grid example under the 2-hop interference model are highlighted in Fig. 9.

We use the indicator function $\mathbf{1}_{\{i \in X\}}$ to indicate whether the flow i passes through the (K, X) -bottleneck, i.e.,

$$\mathbf{1}_{\{i \in X\}} = \begin{cases} 1 & \text{if } i \in X \\ 0 & \text{otherwise.} \end{cases} \quad (\text{III.3})$$

The total flow rate Λ_X crossing the bottleneck X is given by:

$$\Lambda_X = \sum_{i=1}^N \mathbf{1}_{\{i \in X\}}(\lambda_i). \quad (\text{III.4})$$

Let the flow i enter the (K, X) -bottleneck at the node $v_i^{k_i}$ and leave it at the node $v_i^{l_i}$. Hence, $(l_i - k_i)$ equals the number of links in the (K, X) -bottleneck that are used by flow i .

We define λ_X and $A_X(t)$ as follows:

$$\lambda_X = \sum_{i=1}^N \mathbf{1}_{\{i \in X\}}(l_i - k_i)(\lambda_i). \quad (\text{III.5})$$

$$A_X(t) = \sum_{i=1}^N \mathbf{1}_{\{i \in X\}}(l_i - k_i)(A_i(t)). \quad (\text{III.6})$$

B. The reduction technique

In this section, we describe our methodology to derive lower bounds on the average size of the queues corresponding to the flows that pass through a (K, X) -bottleneck.

By definition, the number of links/packets scheduled in the bottleneck, $I_X(t)$ is no more than K , i.e.,

$$\sum_{i=1}^N \mathbf{1}_{\{i \in X\}} \sum_{j=k_i}^{l_i-1} I_i^j(t) = I_X(t) \leq K. \quad (\text{III.7})$$

A flow i may pass through multiple links in X . Among all the flows that pass through X , let F_X denote the maximum number of links in the (K, X) -bottleneck that are used by any single flow, i.e.,

$$F_X = \max_{i=1}^N \mathbf{1}_{\{i \in X\}}(l_i - k_i). \quad (\text{III.8})$$

Let $S_i^k(t)$ denote the sum of queue lengths of the first k queues of flow i at time t , i.e.,

$$S_i^k(t) = \sum_{j=0}^k Q_i^j(t) \quad (\text{III.9})$$

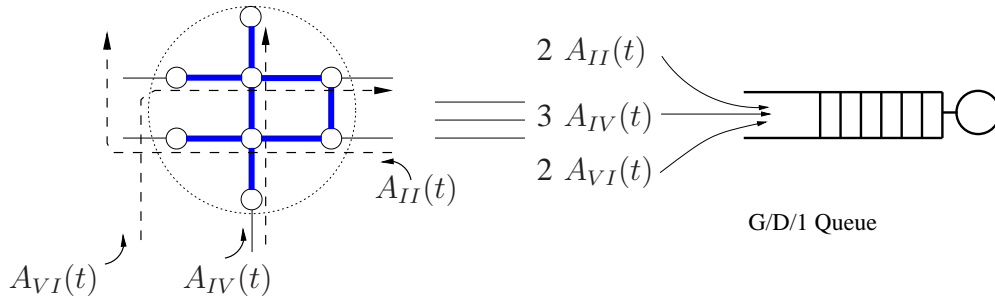


Fig. 2. Reducing a bottleneck exclusive set in Fig. 9 to a G/D/1 queue. Note that $A_{VI}(t)$, $A_{IV}(t)$, $A_{II}(t)$ are external arrivals to the original system, so the arrivals to the reduced G/D/1 system are all external.

Summing Eq. (II.2) from $j = 0$ to k , we have

$$S_i^k(t+1) = S_i^k(t) + A_i(t) - I_i^k(t). \quad (\text{III.10})$$

The sum of queues upstream of each link in X at time t is given by $\mathcal{S}_X(t)$ and satisfies the following property.

$$\begin{aligned} \mathcal{S}_X(t) &= \sum_{i=1}^N \mathbf{1}_{\{i \in X\}} \sum_{j=k_i}^{l_i-1} S_i^j(t) \geq \sum_{i=1}^N \mathbf{1}_{\{i \in X\}} \sum_{j=k_i}^{l_i-1} Q_i^j(t) \\ &\geq \sum_{i=1}^N \mathbf{1}_{\{i \in X\}} \sum_{j=k_i}^{l_i-1} I_i^j(t) = I_X(t). \end{aligned} \quad (\text{III.11})$$

Now we consider the evolution of the queues \mathcal{S}_X under an arbitrary scheduling policy which is given by the following equation.

$$\mathcal{S}_X(t+1) = \mathcal{S}_X(t) - I_X(t) + A_X(t). \quad (\text{III.12})$$

Note: By summing the queues upstream of the bottleneck and defining $\mathcal{S}_X(t)$, we are able to avoid correlation terms among the arrival and service processes in the queue evolution equation of the system (Eq. (III.12)). We obtain a lower bound on the value of $\mathcal{S}_X(t)$ in Theorem 3.1 by studying a reduced system. Using the result of Theorem 3.1, we obtain a lower bound on the expected delay for the flows passing through the bottleneck in Corollary 3.1.

Reduced System: Consider a system with a single server and $A_X(t)$ as the input. The server serves at most K packets from the queue. Let $\mathcal{Q}_X(t)$ be the queue length of this system at time t . The queue evolution of the reduced system is given by the following equation.

$$\mathcal{Q}_X(t+1) = (\mathcal{Q}_X(t) - K)^+ + A_X(t) \quad (\text{III.13})$$

$$\text{where} \quad (x)^+ = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

The reduction procedure is illustrated in Fig. 2 where we have reduced one of the bottlenecks in the grid example shown in Fig. 9. Flows II, IV and VI pass through an exclusive set using two, three and two hops of the exclusive set respectively. The corresponding G/D/1 system is fed by the exogenous arrival streams $2A_{II}(t)$, $3A_{IV}(t)$ and $2A_{VI}(t)$.

Without loss of generality we can assume that both systems are initially empty, i.e., $\mathcal{Q}_X(0) = \mathcal{S}_X(0) = 0$. We now establish that at all times $t \geq 0$, $\mathcal{Q}_X(t)$ is smaller than $\mathcal{S}_X(t)$.

Theorem 3.1: For a (K, X) -bottleneck in the system, at any time $T \geq 0$, the sum of the queue lengths \mathcal{S}_X in X , under any scheduling policy is no smaller than that of the reduced system, i.e., $\mathcal{Q}_X(T) \leq \mathcal{S}_X(T)$.

Proof: We prove the above theorem using the principle of mathematical induction.

Base Case: The theorem holds true for $T = 0$, since the system is initially empty.

Induction hypothesis: Assume that the theorem holds at a time $T = t$, i.e., $\mathcal{Q}_X(t) \leq \mathcal{S}_X(t)$.

Induction Step: The following two cases arise.

Case 1: $\mathcal{Q}_X(t) \geq K$

$$\begin{aligned} \mathcal{Q}_X(t+1) &= \mathcal{Q}_X(t) - K + A_X(t) \\ &\leq \mathcal{S}_X(t) - K + A_X(t) \\ &\leq \mathcal{S}_X(t) - I_X(t) + A_X(t) \\ &= \mathcal{S}_X(t+1). \end{aligned} \quad (\text{III.14})$$

Case 2: $\mathcal{Q}_X(t) < K$.

Using Eq. (III.11), we have the following,

$$\begin{aligned} \mathcal{Q}_X(t+1) &= A_X(t) \\ &\leq \mathcal{S}_X(t) - I_X(t) + A_X(t) \\ &= \mathcal{S}_X(t+1). \end{aligned} \quad (\text{III.15})$$

Hence, the theorem is holds for $T = t + 1$.

Thus by the principle of mathematical induction, the theorem holds for all T . \blacksquare

Remarks

- The above analysis captures the combinatorial interference constraints and reduces the bottleneck to a G/D/K system with appropriate inputs for the purpose of establishing lower bounds. Such a system can be analyzed for a large class of arrival traffic.
- Even when the arrival process is not amenable to analysis, the above reduction can be used to obtain a sample path lower bound via simulation. For example, while evaluating a scheduling algorithm via a trace based simulator, we can feed the arrival trace to the corresponding G/D/K system to obtain a lower bound on its performance. Furthermore, a lower bound on important network wide statistics could also be obtained using the above technique along with the flow partition technique described in Sec. III-D.

- The analysis here is general, and establishes a fundamental lower bound even for the traditional wireline setting.
- We emphasize that $A_X(t)$ can be computed from Eq. (III.6) and considers only the exogenous inputs to the system. Furthermore, the lower bound on the expected delay can be computed using only the statistics of the exogenous arrival process and not their sample paths.

We note that a policy may achieve the above lower bound $Q_X(t)$ on the sum of upstream queues if it schedules the same number of packets as the corresponding G/D/K system in every time-slot. However, this may not always be possible because of the interference caused by other flows in the system. It is also important to note that even if a scheme achieves the lower bound on $S_X(t)$, it does not imply that it would be delay-optimal (i.e., it will minimize the total number of packets in the system at all times). We will provide an example of the clique network in section IV-A. We now discuss the derivation of an explicit lower bound on expected delay of the flows passing through the bottleneck using the above theorem.

C. Bound on Expected Delay

We now present a lower bound on the expected delay of the flows passing through the bottleneck as a simple function of the expected delay of the reduced system. In the analysis, we use Theorem 3.1 to bound the queueing upstream of the bottleneck and a simple bound on the queueing downstream of the bottleneck. Applying Little's law on the complete system, we derive a lower bound on the expected delay of the flows passing through the bottleneck.

Corollary 3.1: Let $\mathbf{E}[\widetilde{D}_X]$ be the expected value of queueing delay for the G/D/1 system with input $A_X(t)$. Further let, $\mathbf{E}[D_X]$ be the expected delay of the flows passing through X .

$$\text{Then } \mathbf{E}[D_X] \geq \frac{\mathbf{E}[\widetilde{D}_X]}{F_X} + \frac{\sum_{i=1}^N \mathbf{1}_{\{i \in X\}} \lambda_i (|P_i| - l_i)}{\Lambda_X}$$

Proof: Let $\mathcal{Q}(t)$ denote the queue length of the G/D/1 system at time t . Theorem 3.1 states that at all times,

$$\sum_{i=1}^N \mathbf{1}_{\{i \in X\}} \sum_{j=k_i}^{l_i-1} S_i^j(t) \geq \mathcal{Q}(t).$$

Since for all $j < l_i - 1$, $S_i^j(t) \leq S_i^{l_i-1}(t)$, thus

$$\sum_{i=1}^N \mathbf{1}_{\{i \in X\}} (l_i - k_i) S_i^{l_i-1}(t) \geq \mathcal{Q}(t).$$

Using Eq. (III.8), it follows that,

$$\sum_{i=1}^N \mathbf{1}_{\{i \in X\}} (F_X) S_i^{l_i-1}(t) \geq \mathcal{Q}(t). \quad (\text{III.16})$$

and hence,

$$\sum_{i=1}^N \mathbf{1}_{\{i \in X\}} \sum_{j=0}^{l_i-1} Q_i^j \geq \frac{\mathcal{Q}(t)}{F_X}. \quad (\text{III.17})$$

After crossing the bottleneck, a packet of flow i has to cross $|P_i| - l_i$ hops. Since the links are of unit capacity, the delay at each of these hops is at least one unit. Thus for all $l_i \leq j < |P_i|$,

$$\mathbf{E}[Q_i^j] \geq \lambda_i. \quad (\text{III.18})$$

Taking expectations on both sides of Eq. (III.17) and using Eq. (III.18), we obtain,

$$\sum_{i=1}^N \mathbf{1}_{\{i \in X\}} \sum_{j=0}^{|P_i|} \mathbf{E}[Q_i^j] \geq \frac{\mathbf{E}[\mathcal{Q}(t)]}{F_X} + \sum_{i=1}^N \mathbf{1}_{\{i \in X\}} (\lambda_i (|P_i| - l_i)). \quad (\text{III.19})$$

Applying Little's law, $\mathbf{E}[D_X] =$

$$\frac{\sum_{i=1}^N \mathbf{1}_{\{i \in X\}} \sum_{j=0}^{l_i-1} \mathbf{E}[Q_i^j]}{\Lambda_X} \geq \frac{\mathbf{E}[\widetilde{D}_X]}{F_X} + \frac{\sum_{i=1}^N \mathbf{1}_{\{i \in X\}} \lambda_i (|P_i| - l_i)}{\Lambda_X}. \quad (\text{III.20})$$

D. Flow Partition

Let Z be the set of flows in the system. Let π be a partition on Z such that each element $p \in \pi$ is a set of flows passing through a common (K_p, X_p) -bottleneck. The expected delay of the flows in p can be lower bounded using Corollary 3.1. A system-wide lower bound on the expected delay of the packets, $\mathbf{E}[D]$, can then be obtained by a straight-forward application of the Little's law.

$$\mathbf{E}[D] \geq \frac{\sum_{p \in \pi} \frac{\lambda_{X_p} \mathbf{E}[\widetilde{D}_{X_p}]}{F_{X_p}} + \sum_{i=1}^N \mathbf{1}_{\{i \in X\}} \lambda_i (|P_i| - l_i)}{\sum_{i=1}^N \lambda_i} \quad (\text{III.21})$$

Our objective is to compute a partition π such that the lower bound on $\mathbf{E}[D]$ can be maximized. The optimal partition can be computed using a dynamic program, but the computation costs can be exponential in the number of flows in the worst case. We now present a greedy algorithm which computes a lower bound on the average delay for a system containing multiple bottlenecks.

Assume that we have pre-computed a list of (K, X) -bottlenecks in the system. Algorithm 1 proceeds by greedily searching for a set of flows $p \subset Z$ and the corresponding (K_p, X_p) -bottleneck that yields the maximum lower bound. The value of the variable *BOUND* is incremented and the flows in p are then removed from Z . The process is repeated until all the flows are removed. Thus, we obtain a decomposition of the wireless network into several single queue systems, and obtain a bound on the expected delay.

Algorithm 1 Greedy Partitioning Algorithm

- 1: $Z \leftarrow \{1, 2, \dots, N\}$
 - 2: $BOUND \leftarrow 0$
 - 3: **repeat**
 - 4: Find the (K, X) -bottleneck which maximizes $\mathbf{E}[D_X]$
 - 5: $BOUND \leftarrow BOUND + \Lambda_X \mathbf{E}[D_X]$
 - 6: $Z \leftarrow Z \setminus i : i \in X$
 - 7: **until** $Z = \Phi$
 - 8: **return** $\frac{BOUND}{\sum_{i=1}^N \lambda_i}$
-

The $(1, X)$ -bottlenecks correspond to cliques in the conflict graph [15]. Let M be the largest number of links that interfere

with a link $l \in L$. The time complexity to compute all the $(1, X)$ -bottlenecks is exponential in M in the worst case. In general, the time complexity to compute all the (K, X) -bottlenecks is exponential. However, in our experiments we find that the number of $(1, X)$ -bottlenecks in graphs with 2-hop interference is much smaller. To reduce the complexity of the problem, we can restrict ourselves to computing bottlenecks around the set of links where the several flows converge.

The lower bound analysis may be loose on account of the following. Firstly, inequality (III.16) is loose when the flows pass through different number of links in the same bottleneck. Secondly, the lower bound obtained by Algorithm 1 can capture the effect of any flow at only one bottleneck. Hence, it would underestimate the congestion caused by a flow passing through multiple bottlenecks. Thirdly, we assume that the queuing in each bottleneck is independent of each other, which may not be possible because of interference among two bottlenecks. Finally, in the derivation of the lower bound by the reduction technique, we have neglected the non-empty queue constraints by grouping the arrivals into a single queue, and hence we underestimate the delay. We evaluate the impact of these relaxations on the accuracy of the lower bound using simulations. Despite these relaxations, we find that the lower bound gives a useful estimate of the average delay in the system.

IV. DESIGN OF DELAY EFFICIENT POLICIES

We now address the important question of designing a delay-efficient scheduler for general multi-hop wireless networks. We will see that although delay optimal policies can be derived for some simple networks like the clique and the tandem, deriving such policies in general is extremely complex. Intuitively, such a scheduler must satisfy the following properties.

- *Ensure high throughput:* This is important because if the scheduling policy does not guarantee high throughput then the delay may become infinite under heavy loading.
- *Allocate resources equitably:* The network resources must be shared among the flows so as not to starve some of the flows. Also, non-interfering links in the network have to be scheduled such that certain links are not starved for service. Starvation leads to an increase in the average delay in the system.

The above properties are difficult to achieve; given the dynamics of the network and the lack of a priori information of the packet arrival process. In the light of the previous work [20], [30], we choose to investigate the *back-pressure policy with fixed routing* (Section IV-B). The back-pressure policy has been widely used to develop solutions for a variety of problems in the context of wireless networks [8], [20]; and the importance of studying the trade-offs in stability, delay, and complexity of these solutions is now being realized by the research community. This policy tries to maintain the queues corresponding to each flow in decreasing order of size from the source to the destination. This is achieved by using the value of differential backlog (difference of backlogs at the two ends of a link) as the weight for the link and scheduling

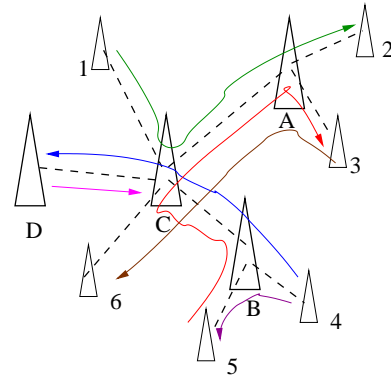


Fig. 3. An example network (clique) with interference constraints such that only one pair of nodes can communicate at any given time. Note that the packets may still need to traverse multiple hops.

the matching with the highest weight. As a result, the policy is throughput optimal. Henceforth, we shall refer to this policy as only the back-pressure policy.

We first study the delay optimal policy for a clique network. We then modify the back-pressure policy using the intuition gained from the nature of the delay optimal scheduling for the clique and tandem networks.

A. Clique

A clique network is one in which the interference constraints allow only one link to be scheduled at any given time. Such a situation may arise, for example, in the down-link of a base station which employs relays to increase coverage and/or data rates (see for eg. [29]). Suppose there are N flows in the clique network. An example network with six flows is shown in Fig. 3. Every link lies in the interference range of the other and hence only one link can be scheduled at any given time.

It is well known [26], [28] that the Shortest Remaining Processing Time (SRPT) policy is delay optimal in a work-conserving queue with preemption. Using this result, we can design a scheduling policy that minimizes the total number of packets in the system at all times for every sequence of arrivals. This is also known as *sample path delay optimality*. In particular, we will show that for the given network, scheduling the packet which is closest to its destination is optimal.

Let h be the maximum number of hops, a flow takes in the clique network.

$$h = \max_{i=1}^N |P_i| \tag{IV.22}$$

For the sake of simplicity, we define a h -dimensional vector $\mathbf{q}(t)$, which represents the state of the system: $\mathbf{q}(t) = (\sum_{i=1}^N Q_i^{|P_i|-1}(t), \sum_{i=1}^N Q_i^{|P_i|-2}(t), \dots, \sum_{i=1}^N Q_i^{|P_i|-h}(t))$, where $Q_i^k = 0 \quad \forall \quad k < 0$.

Note that we do not distinguish packets of different flows in this description of the state. We only consider the distance of a given packet from its destination. Let $q^j(t)$ be the number of packets that are j hops from their respective destinations at time t . The following equation describes the evolution of \mathbf{q}

when the activation vector I^j is scheduled at time t .

$$\begin{aligned} \mathbf{q}(t+1) &= \mathbf{q}(t) - I^j(t) + A(t) \\ \mathbf{q}^j(t+1) &= \mathbf{q}^j(t) - 1 + A^j(t) \\ \mathbf{q}^{j-1}(t+1) &= \mathbf{q}^{j-1}(t) + 1 + A^{j-1}(t) \quad \text{if } j > 1. \end{aligned} \quad (\text{IV.23})$$

A scheduling operation I^j schedules a packet from $\mathbf{q}^j(t)$, provided that the queue is non-empty. $A^j(t)$ is the number of exogenous packets arriving to the system at time t , which are j hops from their respective destinations. The optimal scheduling rule, I^{opt} schedules

$$\min_j : \mathbf{q}^j > 0. \quad (\text{IV.24})$$

We begin with the proof of the sample-path delay optimality result.

Lemma 4.1: Consider the evolution of the system under the policy I^{opt} and an arbitrary policy I . Let $\mathbf{q}(t), \mathbf{q}^{opt}(t)$ be the queue length processes under I and I^{opt} , respectively, when the system starts from the same initial state under both policies. Assume that the number of arrivals in any slot is finite. For all $t = 0, 1, \dots$ we have $l(\mathbf{q}(t)) \geq l(\mathbf{q}^{opt}(t))$.

Proof: The system is pre-emptive in that a different packet may be scheduled in the next slot. The system is work-conserving because after each scheduling decision the number of hops the packet needs to traverse, decreases by 1. We will describe the mapping of the problem to an equivalent work conserving system.

Assume a single queue system with a single server. Each new arrival of a packet corresponding to a flow in the clique network marks the arrival of a new job to the corresponding single queue system. The remaining service time of the job is equal to its distance from the destination. Hence the SRPT policy corresponds to scheduling the packet closest to the destination. In other words, I^{opt} rule is optimal. ■

Note: Lemma 4.1 tells us that the policy I^{opt} is a sample-path delay optimal policy since its optimality does not depend on the nature of the arrival process. We next show that any work-conserving scheduling policy minimizes \mathcal{S}_X for the clique network (see Eq. III.11 of Section III-B) on a sample path basis although it may not be minimize the sum of queue lengths at all times.

Lower Bound Since no more than one link can be scheduled in the clique network, it is a $(1, X)$ -bottleneck. Let us define \mathcal{S}_X for the clique network (see Eq. III.11 of Section III-B). It can be shown that for the clique network,

$$\mathcal{S}_X(t) = \sum_{k=1}^h k \mathbf{q}^k(t) \quad (\text{IV.25})$$

We now show that any work-conserving policy (that schedules a non-empty queue) will achieve the lower bound on \mathcal{S}_X , i.e. $\mathcal{Q}_X(t)$ at all times t . Suppose I^j is the activation vector

scheduled by the policy.

$$\begin{aligned} \mathcal{S}_X(t+1) &= \sum_{k=1}^{j-2} k \mathbf{q}^k(t) + (j-1)(\mathbf{q}^{j-1}(t) + 1) \\ &\quad + j(\mathbf{q}^j(t) - 1) + A_X(t) \\ &= \sum_{k=1}^h k \mathbf{q}^k(t) - 1 + A_X(t) = \mathcal{Q}_X(t+1) \end{aligned} \quad (\text{IV.26})$$

From the above we conclude that a policy which minimizes \mathcal{S}_X may not minimize the sum of queue lengths in the system at all times, neither it is guaranteed to be delay optimal. It is interesting to note that the optimal policies for the clique network and the tandem network (see [31]) have the property that they give priority to the packets, which are closest to their destination. The rationale behind this choice is that such a decision drains out the system in minimum possible time (also called clearance time [10], [11]) which is a necessary (but not sufficient) condition for delay optimality. In the context of stochastic networks, such policies are called the Last Buffer First Serve (LBFS) policies [21]. At the other end of the spectrum are the First Buffer First Serve (FBFS) policies [21] which incur maximum delay among the class of work-conserving policies. We simulate these networks in Section V and observe that the average delay of the optimal policy is close to the lower bound.

However, as shown in [11], it is impossible to design policies that minimize the total number of packets in the system at all times, even for a simple switch (equivalent to a wireless network with bipartite graph and node-exclusive interference). We instead take the approach of modifying the back-pressure policy by increasing the relative priority of packets that are within one hop of their destinations.

B. Back-Pressure Policy

The back-pressure policy may lead to large delays since the backlogs are progressively larger from the destination to the source. The packets are routed only from a longer queue to a shorter queue and certain links may have to remain idle until this condition is met. Hence, it is likely that all the queues upstream of a bottleneck will grow long leading to larger delays. A common observation of the optimal policies for the clique and the tandem network is that increasing the priority of packets which are close to the destination reduces the delay. In the context of wireline (stochastic-processing) networks this is known as the Last Buffer First Serve (LBFS) rule studied in [21]. We introduce a new family of functions parameterized by α for computing the differential backlogs of the links. Using the parameter α , we control relative priority of links. This in turn influences the average delay of the system.

Our simulations indicate that for certain topologies, for an appropriate choice of α , the average delay in the above system can be reduced close to the fundamental lower bound. For a tandem queue network, as α goes to zero, the delay performance of the back-pressure policy numerically coincides with that of the delay optimal policy proposed by [31] and

also the lower bound provided in this paper. We give a formal description of the back-pressure policy.

Let $e := (a, b)$ be a link of interest. Suppose that flow i passes through link e and that nodes a and b are at a distance of j and $j + 1$ hops, respectively, from the source node s_i . In our notation, $e := (v_i^j, v_i^{j+1})$. Define the differential backlog ∇Q_e^i of flow i passing through a link $e := (v_i^j, v_i^{j+1})$ as

$$\nabla Q_e^i = (Q_i^{j_i})^\alpha - (Q_i^{j_i+1})^\alpha, \quad \text{for some } \alpha > 0 \quad (\text{IV.27})$$

For each link e , the flow with the maximum differential backlog is chosen by the flow scheduling component (Eq. (IV.28) in Fig. 4). The link scheduling component shown in Fig. 4 schedules the activation vector with the maximum weight at every time slot. A packet of flow i is transmitted on link $e := (v_i^j, v_i^{j+1})$ at time t if flow i had the maximum differential backlog at link e , link e was present in the maximum weighted matching, and the corresponding queue was non-empty.

Flow Scheduling	
For each link $e \in L$, find the flow with the maximum differential backlog	
$f_e^*(t) = \operatorname{argmax}_i \nabla Q_e^i$	(IV.28)
Assign weights to every link	
$w_e = \max(\nabla Q_{f_e^*}^e, 0)$	(IV.29)
<hr/>	
Link Scheduling	
Schedule the maximum weighted matching	
$\mathbf{I}(t) = \operatorname{argmax}_{\mathbf{J} \in \mathcal{J}} \langle \mathbf{w}, \mathbf{J} \rangle$	(IV.30)
where for two vectors \mathbf{x} and \mathbf{y} , $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_l x_l y_l$ denotes the inner product.	

Fig. 4. Back-Pressure Policy With Fixed Routing

Let $\|\mathbf{Y}\|$ denote the Euclidean norm of vector \mathbf{Y} . The system is considered to be stable if $\lim_{t \rightarrow +\infty} \mathbf{E}[\sup \|\mathbf{Q}(t)\|]$ is bounded. If the system is stable then the throughput of a given flow is the same as the arrival rate. A throughput vector λ is admissible if there is some scheduling policy under which the system is stable when the arrival rate vector is λ . We denote by \bar{C} the closure of the convex hull of the set of activation vectors \mathcal{J} , and by C the interior of the convex hull. Let $\mathbf{1}_{\{i \in e\}}$ be indicator variable indicating whether the flow i passes through link e . The sum of the rates of the flows sharing link e is given by

$$g_e = \sum_{i=1}^N \mathbf{1}_{\{i \in e\}} \lambda_i. \quad (\text{IV.31})$$

Let \mathbf{g} be the corresponding flow rate vector.

It has been shown in [30] that if each arrival process is *i.i.d.* in time, and that the first two moments of all the arrival streams $\{A_i(t)\}_{t=1}^\infty$ are finite, then $\mathbf{g} \in C$ is a necessary condition

for a stabilizing scheduling policy to exist. It has also been shown that the back-pressure policy with fixed routing (with $\alpha = 1$) stabilizes the system for any arrival rate satisfying the preceding condition. It can be shown using the fluid model techniques developed in [4] and [27] that the above policy with ($\alpha > 0$) is stable whenever the arrival processes satisfy a strong-law-of-large numbers assumption and the flow rate vector $\mathbf{g} \in C$.

V. ILLUSTRATIVE EXAMPLES

We now demonstrate our methodology on a variety of examples shown in Fig. 5. The bottleneck sets in each example have been highlighted in the corresponding figures. We also support the lower bound with results obtained from the simulations of the back-pressure policy (Section IV-B) to show that the lower bounds are indeed useful. Not only does the lower bound serve as a rough estimate, but can also be used to gain understanding on the back-pressure policy itself. Further, we also compare the performance of the back-pressure policy with the maximal policy [3], [33] in Sections V-C and V-D.

In all cases, we present the results of one dimensional scaling of the load. Once a base-line load is chosen, we use a scaling factor to study the delay performance for different values of the (normalized) load in the network. The maximum value of the load that is supported by the system is determined from the simulations. Since back-pressure policy is throughput optimal, the point where the system becomes unstable must be outside the capacity region. In these systems, λ is a vector of average arrival rates and $\mathbf{A}(t)$ is the vector of exogenous arrivals to the system as defined in Section II. In this section, we use ρ , a scalar quantity which denotes the utilization of the system, i.e. relative loading with respect to the capacity of the system. The value of ρ corresponds to the point on the boundary of the capacity region.

We implemented an algorithm to compute all the exclusive sets of a graph under a given interference model. We also implement Algorithm 1 and the back-pressure policy described in Section IV-B. Cplex [13], an integer-programming solver, was used to compute the maximum weight matchings. Except for the Tandem Queue, the 2-hop interference model has been used in all other simulations. All the simulations have been run long enough for the 95% confidence intervals to become small as shown in Figs. 8 and 10.

Arrival Processes: The arrival stream at each source is a series of active and idle periods. During the active periods, the source injects one packet into the queue in every time slot. The length of the active periods (denoted by random variable a) are distributed according to the Zipf law with power exponent 1.25 and finite support $[1, 2, 3, \dots, 100]$. Truncated heavy-tailed distributions like Zipf, have been found to model the Internet traffic [7]. During the active period the source generates one packet every time-slot. The idle periods are geometrically distributed with mean p . The mean arrival rate of a source can be controlled by changing the value of p . The lower bounds were obtained using Algorithm 1. We use the analysis in [6] to obtain the expected delay for the single queue systems.

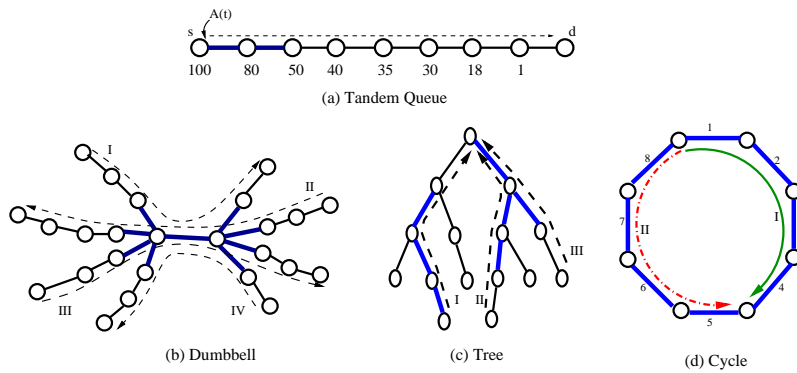


Fig. 5. Illustration of the Lower Bound analysis (bottlenecks used in the analysis have been highlighted).

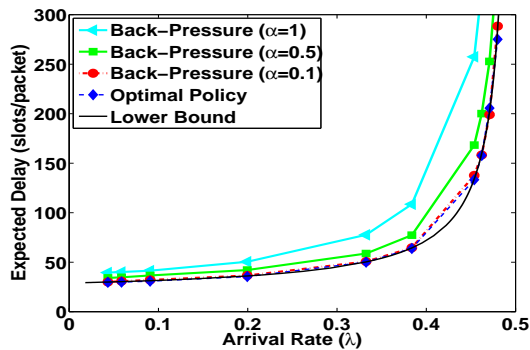


Fig. 6. Simulation results for Tandem Queue

A. Tandem Queue

We consider a stream of packets flowing over the wireless links in tandem, as shown in Fig. 5(a) under the 1-hop interference model. For this system, any two links that are adjacent to each other form an exclusive set. Choosing the first two links as the bottleneck maximizes the lower bound in Corollary 3.1 as it maximizes the value of $(|P_i| - l_i)$. Note that $\frac{\mathbb{E}[D_X]}{F_X}$ is the same for all exclusive sets in the system. The lower bound for the above arrival process is given by,

$$\mathbb{E}[D_t] \geq \frac{2(\zeta - 0.5)\lambda(1 - \lambda)^2}{1 - 2\lambda} - \frac{\lambda^2(1 - \lambda)}{1 - 2\lambda} + \zeta\lambda^2 - \lambda(\zeta - 1) + 6 \quad (\text{V.32})$$

where λ is the arrival rate in packets/slot and $\zeta = \frac{\mathbb{E}[a^2]}{2\mathbb{E}[a]} + 0.5$ is the mean residual time in an active period. Simulation results in Fig. 6 show that this lower bound virtually coincides with the delay performance of the optimal scheme [31].

The value of α in the back-pressure policy can be used to control the relative priority of links. For example, assume that the queue lengths at different nodes in the tandem queue are as shown in Fig. 5(a). For $\alpha = 1$, the differential backlogs ∇Q_1^1 through ∇Q_1^8 are 20, 30, 10, 5, 5, 12, 17 and 1, respectively. For $\alpha = 0.1$, the differential backlogs ∇Q_1^1 through ∇Q_1^8 are 0.035, 0.071, 0.033, 0.019, 0.022, 0.007, 0.335 and 1, respectively. Notice that as the value of α decreases, the value of differential backlog between two non-empty queues becomes smaller. The differential backlog at the last hop becomes comparatively large for small values of α , thereby

increasing the relative priority of the last link. We observe for small values of α , most of the queueing takes place in the first few hops of the flow and the average backlogs at the downstream links are very small. Intuitively, the scheme reduces to the delay optimal scheme as confirmed by the simulation results shown in Fig. 6.

Indeed, the scheme in [31] is valid only for the tandem queue under 1-hop interference model. It has been suggested in the literature [17], [27] that the delay performance of single-hop systems improves as α goes to zero. We also observe a similar pattern for the tandem queue. However, as we will see later, this observation may not be generalizable to a multi-hop wireless networks with several flows, since for small values of α , certain flows may be starved for resources.

B. Clique

We now consider the clique network with six flows as shown in Fig. 3 with the load vector $\rho(0.0625, 0.0625, 0.0625, 0.0625, 0.0625, 0.0625)$ packets/slot. We observe in Fig. 7, that the optimal policy (Last Buffer First Serve) derived in the previous section is indeed closest to the lower bound. The gap between the performance of LBFS and the lower bound can be attributed to the fact that the flows differ in path length, making inequality (III.16) loose. On the other hand we notice that the First Buffer First Serve (FBFS) policy incurs the maximum delay among the policies simulated herein. Note that for this example, every work conserving policy is throughput optimal and also minimizes $S_X(t)$ at all times. However the average delay performance is dependent on the relative priority of the buffers. It is evident from Fig. 7 that decreasing the value of α in the back-pressure policy improves its delay performance.

C. Dumbbell topology

Consider a dumbbell topology (Fig. 5(b)), with multiple flows passing through a single link with the load vector $\rho(0.1, 0.2, 0.15, 0.05)$ packets/slot. The bottleneck exclusive set has been highlighted in Fig. 5(b). The performance of back-pressure policy ($\alpha = 0.1$) is compared with that of the lower bound in Fig. 8. This example shows that the back-pressure policy is able to share the resources among the flows in a manner such that the overall delay of the system is close to

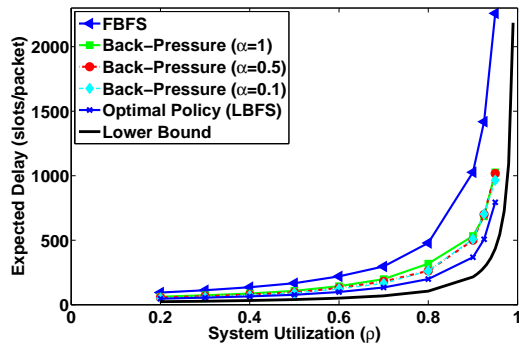


Fig. 7. Simulation results for Clique shown in Fig. 3

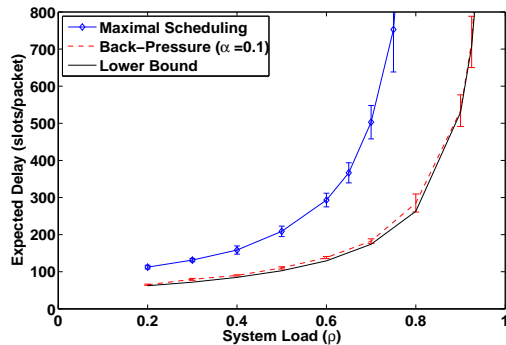


Fig. 8. Simulation results for Dumbbell Topology

the lower bound. We also note that the delay performance of the back-pressure policy is significantly better than that of the maximal scheduling policy [3], [33].

D. Tree Topology

Consider a tree topology with three flows converging at the root of the tree shown in Fig. 5(c) with load vector $\rho(0.2, 0.25, 0.15)$ packets/slot. Such a topology is often found in sensor networks, wireless access networks etc. For the case simulated here, Algorithm 1 decomposes the system into two bottlenecks. Note that the inequality (III.16) would be loose, since flows II and III pass through different number of links in the bottleneck. Also, Algorithm 1 underestimates the lower bound by neglecting the interference between the two bottlenecks. As shown in Fig. 10, the performance of back-pressure policy ($\alpha = 0.1$) is significantly better than that of the maximal scheduling policy and is also close to the lower bound. This suggests that the impact of the relaxations made in the analysis is relatively small in this case.

E. Cycle Topology

We now illustrate the application of the lower bound analysis to a $(2, X)$ -bottleneck. For the given cycle network in Fig. 5(d), no more than two links can be scheduled at any time under 2-hop interference constraints, i.e. for $X = \{1, 2, 3, \dots, 8\}$, $I_X(t) \leq 2$. It can be easily verified that the analysis presented in Section III-B can be used to reduce the

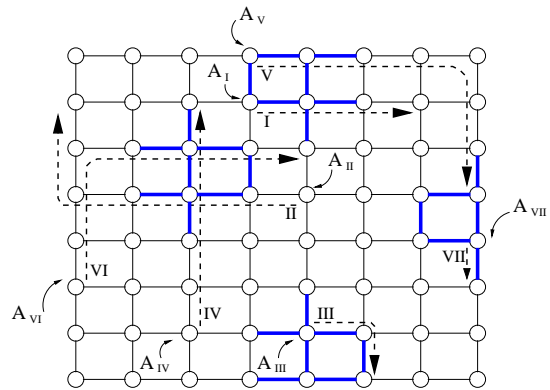


Fig. 9. A grid network with multiple flows. Some of the important bottlenecks have been highlighted.

system to a $G/D/2$ queue having arrivals $4A_I(t)$ and $4A_{II}(t)$ respectively.

We simulated the system with load vector $\rho(0.25, 0.25)$ packets/slot and observed that the lower bounds derived by the analysis of the $G/D/2$ system are tighter when $\rho > 0.7$. We also obtained lower bounds using the bottlenecks corresponding to exclusive sets $\{1, 2, 3\}$ and $\{6, 7, 8\}$ for flows I and II respectively. These bounds are nonetheless useful for light loads. Thus, it is possible to derive accurate lower bounds for the wireless system by considering appropriate bottlenecks.

We would like to note that in this case, flows I and II have the same source and destination nodes and the two flows interact closely with each other because of link interference. We found that for $\alpha \leq 0.1$, flow II is starved when the system is highly loaded ($\rho > 0.9$). This is because, for small values of α , the differential backlogs are very similar in value to each other, even if some of the queues are very long. Hence, the algorithm is not able to preferentially schedule the longer queues in the system. For $\alpha = 0.1$, the performance at lighter loads is however, very similar to that for $\alpha = 0.25$, which is shown in Fig. 11.

F. Analysis of the Example in Fig. 9

In this example, we analyze the wireless grid with randomly generated flows described in Fig. 9. There are several bottlenecks in this system which interfere with each other under the 2-hop interference model. We studied the system for several different input load vectors λ . We find that depending on the input load vector, Algorithm 1 computes different partitions for the flows in the system. We discuss two representative load vectors to evaluate the impact of the relaxations made in the analysis.

Case 1:

$\lambda = (0.12, 0.15, 0.12, 0.06, 0.12, 0.15, 0.12)$ packets/slot. For the given load vector, Algorithm 1 computes the partition $\{\{II, VI\}; \{I, V\}; \{III\}; \{IV\}; \{VII\}\}$. Note that flow IV interferes with flows II and VI significantly, but this effect is not captured by the lower bound analysis. Also note that flow I interferes with flows IV and VI. The lower bound computed by Algorithm 1 is 185.9 slots/packet. We also simulated the system under the back-pressure policy for several different

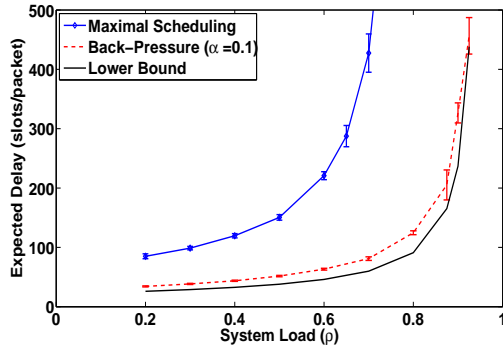


Fig. 10. Simulation results for Tree Topology

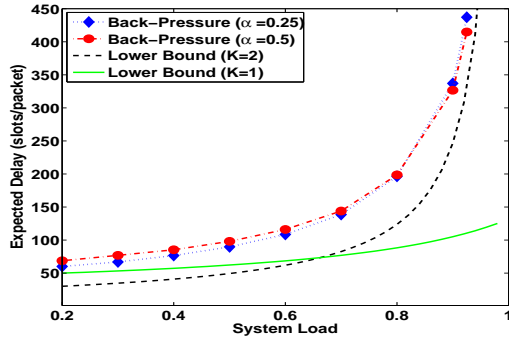


Fig. 11. Simulation results for Cycle Topology

values of α . The average delay was found to be 308.0 slots/packet for $\alpha = 0.4$. For smaller values of α , flow VI was starved for resources resulting in larger delays. The average delay for $\alpha = 0.25$ and $\alpha = 0.1$ was 323.3 slots/packet and 476.2 slots/packet respectively.

Case 2:

$\lambda = (0.12, 0.15, 0.12, 0.0, 0.12, 0.15, 0.12)$ packets/slot. In this case, we remove flow IV from the system and keep all other arrivals rate the same. The lower bound computed by Algorithm 1 is 196.0 slots/packet. The average delay under the back-pressure policy was found to be 230.7 slots/packet for $\alpha = 0.25$ which is in better agreement with the lower bound as compared to the previous case, even though flow I interferes with flow VI. Interestingly, in this case, decreasing the value of α causes an increase in the queues along flow II, while increasing the value of α causes an increase in the queues along flow VI. The average delay for $\alpha = 0.4$ and $\alpha = 0.1$ was 254.7 slots/packet and 244.1 slots/packet respectively.

These examples also show that it is non-trivial to predict the value of α in the back-pressure policy that minimizes the average delay in the system. Thus, small value of α is not sufficient for the policy to be delay-efficient.

G. Linear Network

Finally, we discuss an example of a line network with several “short” flows and a single “long” flow as shown in Fig. 12. The packet arrival rate (λ) is the same for all the flows

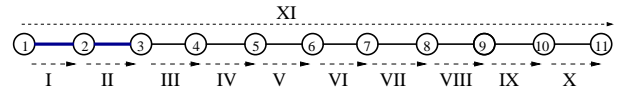


Fig. 12. A linear network with multiple short flows and a single long flow.

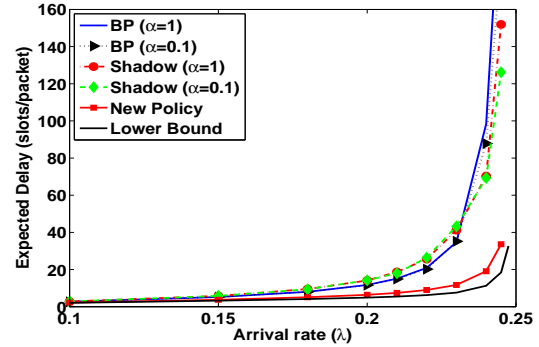


Fig. 13. Simulation results for the Linear Network

in this network. We use a 1-hop interference model. In this example, it is very difficult to allocate the resources equitably and at the same time reduce the backlogs in the system. For example, by using a small value of α the short flows (I through X) get much higher priority in comparison to the flow XI. On the other hand, for a large value of α , the backlogs upstream of flow (in the case of flow XI) are large. Hence, it is not possible to reduce the delay simply by changing α as indicated by Fig. 13. We then implement the *Shadow* scheme proposed in [2] to alleviate the problem of large backlogs associated with back-pressure algorithm by using counters called shadow queues to allocate service rates to each flow on each link in an adaptive fashion without knowing the set of packet arrival rates. However, we find that it does not reduce the queueing in the system. Comparing the performance of these algorithms with the lower bound in Fig. 13, we inferred that there must be policies that incur smaller delay in the system. We then design a new scheduling policy, which although is not guaranteed to be optimal, has much better delay performance. In fact, its performance is close to the lower bound as shown in Fig. 13.

The scheme *New Policy* is based on the observation that the packet closer to the destination must be given higher priority. We implement the scheduling rule followed by Tassiulas’ optimal policy in [31]. Thus, we schedule links beginning from last link (10, 11) and go up to the first link (1, 2). Note that the packets do not have a common destination. Thus, once the link schedule is obtained, we schedule the flow on the link for which the packet is closest to its destination; i.e. we schedule the short flow in preference to the long flow.

We conclude that the lower bound analysis presented here can play an important role in obtaining insights into the design and evaluation of scheduling policies for multi-hop wireless networks. The following section provides a perspective on the research on delay analysis in multi-hop wireless networks and the contributions made in this paper.

VI. DISCUSSION AND RELATED WORK

Much of the analysis [3], [8], [20] for multi-hop wireless networks has been limited to establishing the stability of the system. Whenever there exists a scheme that can stabilize the system for a given load, the back-pressure policy is also guaranteed to keep the system stable. Hence, it is referred to as a throughput-optimal policy. It also has the advantage of being a myopic policy in that it does not require knowledge of the arrival process. In this paper, we have taken an important step towards the expected delay analysis of these systems.

The general research on the delay analysis of scheduling policies has progressed in the following main directions:

- *Heavy traffic regime using fluid models:* Fluid models have typically been used to either establish the stability of the system or to study the workload process in the heavy traffic regime. It has been shown in [5] that the maximum-pressure policy (similar to the back-pressure policy) minimizes the workload process for a stochastic processing network in the heavy traffic regime when processor splitting is allowed.
- *Stochastic Bounds using Lyapunov drifts:* This method is developed in [8], [19], [23], [24] and is used to derive upper bounds on the average queue length for these systems. However, these results are order results and provide only a limited characterization of the delay of the system. For example, it has been shown in [24] that the maximal matching policies achieve $\mathcal{O}(1)$ delay for networks with single-hop traffic when the input load is in the reduced capacity region. This analysis however, has not been extended to the multi-hop traffic case, because of the lack of an analogous Lyapunov function for the back-pressure policy.
- *Large Deviations:* Large deviation results for cellular and multi-hop systems with single hop traffic have been obtained in [32], [35] to estimate the decay rate of the queue-overflow probability. Similar analysis is much more difficult for the multi-hop wireless network considered here, due to the complex interactions between the arrival, service, and backlog process.

Traditional heavy traffic results have focused on a single bottleneck in the system [18], [22], [25] and proving a state-space collapse. We have shown in [9] (Section 4.3.1) that in general, it is impossible to avoid *idling* in these systems. Hence, it may not even be possible to prove a state-space collapse except for some special scenarios. We on the other hand, analyze queueing in multiple bottlenecks by relaxing the constraints in the system, which is crucial to obtain tight lower bounds. Although it may not be possible to achieve the lower bound, our relaxation approach is novel and leads to non-trivial lower bounds.

Here, we have taken a different approach to reduce the wireless network to single queueing systems which are then analyzed to construct the lower bound. We had demonstrated the usefulness of the lower bound for a wireless network with single hop traffic in [12]. It was also observed that the upper bound obtained via the method of Lyapunov drifts was much loose in comparison to the lower bound. The case of multi-

hop flows is technically very challenging and although several results in the literature are available for the delay analysis of networks with single-hop flows, it has been extremely difficult to generalize them for the case of multi-hop flows. In this paper, we develop techniques to characterize the behavior of the system in terms of the external arrivals by the use of our "reduction technique" and overcome the complex interactions in the network. We show that our analysis captures the essential features of the wireless network and is useful since, in many cases, we can design a policy that performs close to the lower bound. Perhaps, the most important advantage of the lower bound is that it can be used for analyzing a large class of arrival processes using known results in the queueing literature [6].

Our approach, however, depends on the efficient computation of the bottlenecks in the system. A complete characterization of the bottlenecks in a multi-hop wireless network is an extremely difficult problem. Exclusive sets characterized in [12], [15] prove to be a good beginning for delay analysis. However, they are not enough to obtain tight lower bounds, as shown in the case of a cyclic network.

The design of a delay optimal policy that achieves minimum possible average delay of packets in the network for a given routing matrix has proved to be very challenging. Except for a delay optimal scheduling scheme for the tandem queueing network under the node exclusive interference model derived in [31] and small switches [16], no result is known for other topologies and interference models.

In [34], delay optimal schemes for wireless networks have been proposed, which typically minimize an expected delay metric assuming that the system behaves as M/M/1. Given the complexity involved in scheduling link transmissions in a multi-hop wireless system, the M/M/1 approximation is too coarse.

In [14], the authors propose a policy that guarantees that the per-flow end-end packet delay is within a constant factor of the optimal, for a node-exclusive interference model, whenever the input load, λ , is within 1/5 of the capacity region, C . The analysis is carried out for Poisson traffic using Kelly's theorem for quasi-reversible networks. The "poissonation" scheme mentioned in Section 9 of the paper can, however, cause the delay of the system to grow substantially. The lower bound analysis presented in this paper is applicable for an arbitrary $\lambda \in C$, a more general class of arrival processes and more general interference constraints.

The MWM- α algorithm was studied for switches by [17] and their simulations suggest that the delay of the system reduces with the value of alpha. This algorithm was also analyzed in the heavy traffic regime using fluid models for the case of switched networks (that include single-hop wireless networks) by [27] and it was conjectured that the delay of the system reduces as α goes to zero. However neither of these studies focused on multi-hop wireless networks. It is also not clear that the multiplicative state-space collapse observed for the MWM- α policy for switched networks will also be observed with the back-pressure policy for multi-hop wireless networks. As noted by us in the discussions in Sections V-E and V-F, some of the flows may be starved for resources when

α is small. Hence the intuition from the single-hop case does not automatically generalize to the multi-hop case.

VII. CONCLUSION

The delay analysis of wireless networks is largely an open problem. In fact, even in the wireline setting, obtaining analytical results on the delay beyond the product form types of networks has posed great challenges. These are further exacerbated in the wireless setting due to complexity of scheduling needed to mitigate interference. Thus, new approaches are required to address the delay problem in multi-hop wireless systems. To this end, we develop a new approach to reduce the bottlenecks in a multi-hop wireless to single queue systems to carry out lower bound analysis.

For a special class of wireless systems (cliques), we are able to apply known techniques to obtain a sample path delay-optimal scheduling policy. We also obtain policies that minimize a function of queue lengths at all times on a sample path basis. Further, for a tandem queueing system, we show numerically that the expected delay of a previously known delay-optimal policy coincides with the lower bound.

The analysis is very general and admits a large class of arrival processes. Also, the analysis can be readily extended to handle channel variations. The main difficulty, however is in identifying the bottlenecks in the system. The lower bound not only helps us identify near-optimal policies, but may also help in the design of a delay-efficient policy as indicated by the numerical studies.

ACKNOWLEDGMENTS

The authors would like to thank the reviewers for their comments to improve the presentation of the paper. In particular, one reviewer brought to our attention, the results on SRPT policies and suggesting a mapping to the optimal policy for the clique network.

REFERENCES

- [1] H. Balakrishnan, C. Barrett, V. Kumar, M. Marathe, and S. Thite. The distance-2 matching problem and its relationship to the maclayer capacity of ad hoc networks. *IEEE Journal on Selected Area in Communications*, 22, 2004.
- [2] L. Bui, R. Srikant, and A. L. Stolyar. Novel architectures and algorithms for delay reduction in back-pressure scheduling and routing. *INFOCOM Mini-Conference*, 2009.
- [3] P. Chaporkar, K. Kar, and S. Sarkar. Throughput guarantees through maximal scheduling in wireless networks. In *43rd Annual Allerton Conference on Communication, Control, and Computing*, 2005.
- [4] J. G. Dai and W. Lin. Maximum pressure policies in stochastic processing networks. *Operations Research*, 53:197–218, 2005.
- [5] J. G. Dai and W. Lin. Asymptotic optimality of maximum pressure policies in stochastic processing networks. Preprint, October 2007.
- [6] H. Dupuis and B. Hajek. A simple formula for mean multiplexing delay for independent regenerative sources. *Queueing Systems Theory and Applications*, 16:195–239, 1994.
- [7] A. Feldmann, N. Kammenhuber, O. Maennel, B. Maggs, R. D. Prisco, and R. Sundaram. A methodology for estimating interdomain web traffic demand. In *IMC*, 2004.
- [8] L. Georgiadis, M. J. Neely, and L. Tassiulas. *Resource Allocation and Cross-Layer Control in Wireless Networks, Foundations and Trends in Networking*, volume 1. Now Publishers, 2006.
- [9] G. R. Gupta. *Delay Efficient Control Policies for Wireless Networks*. Ph.D. Dissertation, Purdue University, 2009.
- [10] G. R. Gupta, S. Sanghavi, and N. B. Shroff. Node weighted scheduling. *SIGMETRICS-Performance'09*, June 2009.
- [11] G. R. Gupta, S. Sanghavi, and N. B. Shroff. Workload optimality in switches without arrivals. *MAThematical performance Modeling and Analysis Workshop*, June 2009.
- [12] G. R. Gupta and N. B. Shroff. Delay analysis for wireless networks with single hop traffic and general interference constraints. *IEEE Transactions on Networking*, 18:393 – 405, April 2010.
- [13] I. Ilog. Solver cplex, 2007. <http://www.ilog.com/products/cplex/>.
- [14] S. Jagabathula and D. Shah. Optimal delay scheduling in networks with arbitrary constraints. In *ACM SIGMETRIC/Performance*, June 2008.
- [15] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. In *MOBICOM*, 2003.
- [16] T. Ji, E. Athanasopoulou, and R. Srikant. Counter-examples to the optimality of mws-alpha policies for scheduling in generalized switches. *preprint*, 2009.
- [17] I. Keslassy and N. McKeown. Analysis of scheduling algorithms that provide 100% throughput in input-queued switches. In *39th Annual Allerton Conference on Communication, Control, and Computing. Monticello, Illinois*, October 2001.
- [18] C. N. Laws. *Dynamic Routing In Queueing Networks*. Ph.D. Dissertation, Christ's College, University of Cambridge, 1990.
- [19] E. Leonardi, M. Mellia, F. Neri, and M. A. Marsan. On the stability of input-queued switches with speed-up. *IEEE/ACM Transactions on Networking*, 9(1), February 2001.
- [20] X. Lin, N. B. Shroff, and R. Srikant. A tutorial on cross-layer optimization in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24(8):1452– 1463, Aug 2006.
- [21] S. H. Lu and P. R. Kumar. Distributed scheduling based on due dates and buffer priorities. *IEEE Transactions on Automatic Control*, pages 1406–1416, 1991.
- [22] S. Meyn. *Control Techniques for Complex Networks*. Cambridge University Press, New York, NY, USA, 2008.
- [23] M. J. Neely. Order optimal delay for opportunistic scheduling in multi-user wireless uplinks and downlinks. In *44th Annual Allerton Conference on Communication, Control, and Computing*, September 2006.
- [24] M. J. Neely. Delay analysis for maximal scheduling in wireless networks with bursty traffic. *IEEE INFOCOM*, 2008.
- [25] M. I. Reiman. Some diffusion approximations with state space collapse. *Proceedings of the International Seminar on Modelling and Performance Evaluation Methodology*, pages 209–240, 1983.
- [26] L. Schrage. A proof of the optimality of the shortest remaining processing time discipline. *Operations Research*, 16:687–690, 1968.
- [27] D. Shah and D. Wischik. Heavy traffic analysis of optimal scheduling algorithms for switched networks. Preprint, 2008.
- [28] D. R. Smith. A new proof of the optimality of the shortest remaining processing time discipline. *Operations Research*, 26:197–199, 1978.
- [29] K. Sundaresan and S. Rangarajan. On exploiting diversity and spatial reuse in relay-enabled wireless networks. In *International Symposium on Mobile Ad Hoc Networking and Computing*, 2008.
- [30] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Trans. Aut. Contr.*37, 37(12):1936–1948, 1992.
- [31] L. Tassiulas and A. Ephremides. Dynamic scheduling for minimum delay in tandem and parallel constrained queueing models. *Annals of Operation Research*, Vol. 48, 333-355, 1993.
- [32] V. J. Venkataramanan and X. Lin. Structural properties of ldp for queue-length based wireless scheduling algorithms. In *45th Annual Allerton Conference on Communication, Control and Computing*, 2007.
- [33] T. Weller and B. Hajek. Scheduling nonuniform traffic in a packet-switching system with small propagation delay. *IEEE/ACM Trans. Netw.*, 5(6):813–823, 1997.
- [34] Y. Xi and E. M. Yeh. Optimal capacity allocation, routing, and congestion control in wireless networks. In *International Symposium on Information Theory*, July 2006.
- [35] L. Ying, R. Srikant, A. Eryilmaz, and G. E. Dullerud. A large deviations analysis of scheduling in wireless networks. *IEEE Transactions on Information Theory*, 52:5088–5098, 2006.