Delay Analysis for Multi-hop Wireless Networks

Gagan Raj Gupta
School of Electrical and Computer Engineering
Purdue University, West Lafayette, IN, USA
grgupta@purdue.edu

Ness Shroff
Departments of ECE and Computer Science
The Ohio State University, CO, USA
shroff@ece.osu.edu

***Abstract*—We analyze the delay performance of a multi-hop wireless network with a fixed route between each source-destination pair. There are arbitrary interference constraints on the set of links that can be served simultaneously at any given time. These interference constraints impose a fundamental lower bound on the delay performance of any scheduling policy for the system. We present a methodology to derive such lower bounds. 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. We conduct extensive numerical studies to suggest that the average delay of the back-pressure scheduling policy can be made close to the lower bound by using appropriate functions of queue length.**

I. Introduction

A large number of studies on multi-hop wireless networks have been devoted to system stability and throughput maximization. The delay performance of these systems, however, has largely been an open problem. This problem is enormously difficult even in the context of wireline networks, primarily because of complex network interactions that complicate the queueing mechanisms. In addition, the problem of mutual interference among the links in wireless networks complicates, both, the scheduling mechanisms and their analysis. We present a new, systematic methodology to obtain a lower bound on the system-wide average delay of a packet from source to the destination. Furthermore, we re-engineer a well known scheduling policy to achieve good delay performance vis-a-vis 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 traverse through the network until they reach the destination. 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 cause mutual interference. The set of links that do not cause mutual interference, hence can be scheduled simultaneously, are called activation vectors (matchings). We assume that the set of allowed activation vectors can be arbitrarily chosen, i.e. they can characterize any interference model. For example, a multi-hop wireless grid network with several randomly generated flows is shown in Fig. 1. We refer to this figure throughout the paper, for the purpose of illustration.

The delay performance of any scheduling policy is primarily limited by the interference, which causes many bottlenecks to

This work was supported in part by ARO W911NF-08-1-0238 and NSF award 07221236-CNS.

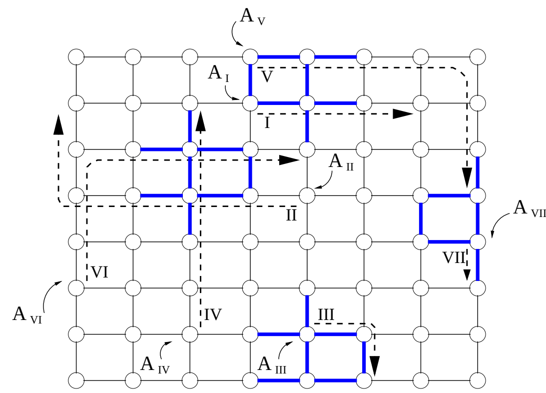

Fig. 1. A typical multi-hop wireless network with multiple flows, each having exogenous arrivals at the source. Some of the important bottlenecks have been highlighted.

be formed in the network. We generalize the typical notion of a 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. We develop an efficient technique to reduce such bottlenecks to a single queue system fed by appropriate arrival processes which are simple functions of the exogenous arrival processes of the original network. The lower bound on the system-wide average delay of a packet is then computed by the analysis of these reduced systems and requires only the statistics of the exogenous arrival processes. For example, [10] characterizes a class of bottlenecks where no more than one link can be scheduled simultaneously. Some of these bottlenecks have been highlighted in Fig. 1.

It is also possible to derive stochastic upper bounds on the average delay of the network using the techniques in [7]. However, we do not pursue them here because they do not capture the effect of statistical multiplexing in the network, which is a critical element of packet switched networks. As a result, these upper bounds tend to be quite loose in most practical scenarios. The lower bound technique on the other hand, captures the effect of interference and statistical multiplexing of packets in the system. Hence, we focus on developing a methodology to establish fundamental lower bounds on the delay performance of any scheduling policy. We consider the lower bound analysis as an important first step towards a complete delay analysis of multi-hop wireless systems.

A delay-efficient 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.

In the light of the previous work [12], [15], we choose the *back-pressure policy with fixed routing (Section II-A)* as a suitable candidate for a delay-efficient scheduler. The back-pressure policy has been widely used to develop solutions for a variety of problems in the context of wireless networks [7], [15]; and the importance of studying the trade-offs in stability, delay, and complexity of these solutions is now being realized by the research community. Under this policy, the queues corresponding to each flow are maintained in decreasing order from the source to the destination. The policy tries to decrease the differential backlog (difference in the backlogs) across each link, by selecting the flow with the highest differential backlog. This value is used as the weight for the link and the matching with the highest weight is scheduled. As a result, the policy is throughput optimal and the resources also tend to be shared equitably among the various flows in the network. Henceforth, we shall refer to this policy as only the 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. Interestingly, we find that by appropriately defining the differential backlogs using a parameter α, the relative priority of links can be controlled. 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 [19] and also the lower bound provided in this paper.

We now summarize our main contributions in this paper:

- Development of a reduction technique to reduce the (K, X)-bottlenecks in the network to single queue systems.
- 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 to suggest that the average delay of the back-pressure scheduling policy can be made close to the lower bound by using appropriate functions of queue length.

We begin with the description of the system model and a brief explanation of the back-pressure policy. We then present our methodology for obtaining reductions and using them to lower bound the system-wide average delay of packets. 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 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 $\mathbf{A}(t) = (A_1(t), \ldots, A_N(t))$ 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 $i \in 1, \ldots, N$). Let $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_N)$ represent the corresponding arrival rate vector.

The path on which flow i is routed is specified as $P_i := (v_i^0, v_i^1, \ldots, v_i^j, \ldots, v_i^{|P_i|})$, 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 $\mathbf{Q}(t) = (Q_i^j(t) : i = 1, 2, \ldots, N)$. 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 $(j+1)^{st}$ hop from source s_i at time slot t. Note that

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

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

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

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 **I**(t) are always separated by h or more hops in the underlying network graph.

A. 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

$$\nabla Q_i^e = (Q_i^{j_i})^\alpha - (Q_i^{j_i+1})^\alpha, \qquad \text{for some } \alpha > 0 \tag{II.3}$$

For each link e, the flow with the maximum differential backlog is chosen by the flow scheduling component (Eq. (II.4) in Fig. 2). The link scheduling component shown in Fig. 2 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

$$f_e^*(t) = \operatorname*{argmax}_i \nabla Q_i^e \tag{II.4}$$

Assign weights to every link

$$w_e = \max(\nabla Q_{f_e^*}^e, 0) \tag{II.5}$$

Link Scheduling

Schedule the maximum weighted matching

$$\mathbf{I}(t) = \operatorname*{argmax}_{\mathbf{J} \in \mathcal{J}} \langle \mathbf{w}, \mathbf{J} \rangle \tag{II.6}$$

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. 2. 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 $\lim_{t \to +\infty} \mathbf{E}[\sup \|\mathbf{Q}(t)\|]$ is bounded. If the system is stable then the throughput of a given flow is the same as the arrival rate. A throughput vector $\boldsymbol{\lambda}$ is admissible if there is some scheduling policy under which the system is stable when the arrival rate vector is $\boldsymbol{\lambda}$. We denote by Λ 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 the 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

$$g_e = \sum_{i=1}^{N} \mathbf{1}_{\{i \in e\}} \lambda_i. \tag{II.7}$$

Let $\mathbf{g}$ be the corresponding flow rate vector. It has been shown in [12] that if each arrival process is *i.i.d.* in time, and that the first two moments of all the arrival streams $\{A_l(t)\}_{t=1}^{\infty}$ 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 [3] and [18] 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 $\mathbf{g} \in C$.

III. Deriving Lower Bounds on Average delay

In this section, we present our methodology to derive lower bounds on the average packet delay for a given multi-hop wireless network. The first step is to identify 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 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. Define a (K, X)-bottleneck to be be a set of links $X \subset L$ such that no more than K of its links can be scheduled simultaneously. For example, [10] identifies 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, 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 were highlighted in Fig. 1.

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

$$\lambda_X = \sum_{i=1}^{N} \mathbf{1}_{\{i \in X\}} (l_i - k_i)(\lambda_i). \tag{III.8}$$

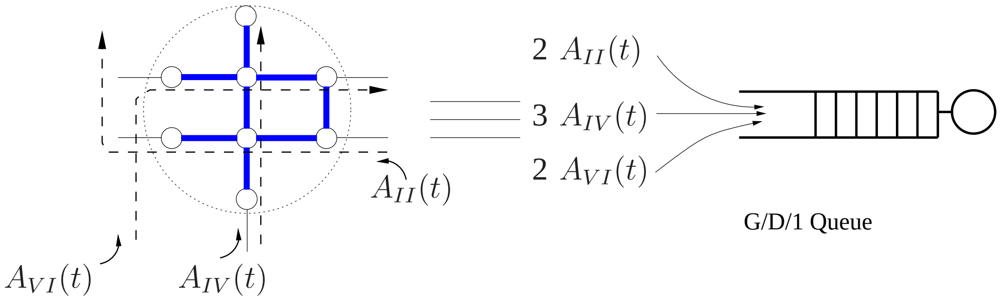

Fig. 3. Reducing a bottleneck exclusive set in Fig. 1 to a G/D/1 queue. Note that $A_{VI}(t)$, $A_{IV}(t)$, $A_{II}(t)$ are external arrivals to the original system, so the arrivals to the reduced G/D/1 system are all external.

$$A_X(t) = \sum_{i=1}^{N} \mathbf{1}_{\{i\in X\}}(l_i - k_i)(A_i(t)). \tag{III.9}$$

B. The reduction technique

In this section, we demonstrate 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.,

$$\sum_{i=1}^{N} \mathbf{1}_{\{i\in X\}} \sum_{j=k_i}^{l_i-1} I_i^j(t) = I_X(t) \leq K. \tag{III.10}$$

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.

$$F_X = \max_{i=1}^{N} \mathbf{1}_{\{i\in X\}}(l_i - k_i). \tag{III.11}$$

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

$$S_i^k = \sum_{j=0}^{k} Q_i^j \tag{III.12}$$

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

$$S_i^k(t+1) = S_i^k(t) + A_i(t) - I_i^k(t). \tag{III.13}$$

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

$$\begin{aligned}\boldsymbol{\mathcal{S}}_X(t) &= \sum_{i=1}^{N} \mathbf{1}_{\{i\in X\}} \sum_{j=k_i}^{l_i-1} S_i^j(t) \geq \sum_{i=1}^{N} \mathbf{1}_{\{i\in X\}} \sum_{j=k_i}^{l_i-1} Q_i^j(t) \\ &\geq \sum_{i=1}^{N} \mathbf{1}_{\{i\in X\}} \sum_{j=k_i}^{l_i-1} I_i^j(t) = I_X(t).\end{aligned} \tag{III.14}$$

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

$$\boldsymbol{\mathcal{S}}_X(t+1) = \boldsymbol{\mathcal{S}}_X(t) - I_X(t) + A_X(t). \tag{III.15}$$

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.15)). We obtain a lower bound on the value of $\boldsymbol{\mathcal{S}}_X(t)$ 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 $\boldsymbol{\mathcal{Q}}_X(t)$ be the queue length of this system at time t. The queue evolution of the reduced system is given by the following equation.

$$\boldsymbol{\mathcal{Q}}_X(t+1) = (\boldsymbol{\mathcal{Q}}_X(t) - K)^+ + A_X(t) \tag{III.16}$$

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

The reduction procedure is illustrated in Fig. 3 where we have reduced one of the bottlenecks in the grid example shown in Fig. 1. 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 empty initially, i.e., $\boldsymbol{\mathcal{Q}}_X(0) = \boldsymbol{\mathcal{S}}_X(0) = 0$. We now establish that at all times t, $\boldsymbol{\mathcal{Q}}_X(t)$ is smaller than $\boldsymbol{\mathcal{S}}_X(t)$.

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

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., $\boldsymbol{\mathcal{Q}}_X(t) \leq \boldsymbol{\mathcal{S}}_X(t)$.

Induction Step The following two cases arise.

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

$$\begin{aligned}\mathcal{Q}_X(t+1) &= \mathcal{Q}_X(t) - K + A_X(t)\\ &\le \mathcal{S}_X(t) - K + A_X(t)\\ &\le \mathcal{S}_X(t) - I_X(t) + A_X(t)\\ &= \mathcal{S}_X(t+1).\end{aligned} \quad \text{(III.17)}$$

Case 2: $\mathcal{Q}_X(t) < K$.
Using Eq. (III.14), we have the following,

$$\begin{aligned}\mathcal{Q}_X(t+1) &= A_X(t)\\ &\le \mathcal{S}_X(t) - I_X(t) + A_X(t)\\ &= \mathcal{S}_X(t+1).\end{aligned} \quad \text{(III.18)}$$

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

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

Remarks

- The above analysis captures the combinatorial interference constraints and reduces the bottleneck to a G/D/1 system with appropriate inputs for the purpose of establishing lower bounds.
- The analysis here is very 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.9) 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 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.

Corollary 3.1: Let $\mathbf{E}[\widetilde{D_X}]$ be the expected value of queuing 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.

Then $\mathbf{E}[D_X] \ge \dfrac{\mathbf{E}[\widetilde{D_X}]}{F_X} + \dfrac{\sum_{i=1}^{N} \mathbf{1}_{\{i \in X\}} \lambda_i(|P_i| - l_i)}{\lambda_X}$

Proof: Let $\boldsymbol{\mathcal{Q}(t)}$ denote the queue length of the G/D/1 system at time t. Theorem 3.1 states that at all times,
$\sum_{i=1}^{N} \mathbf{1}_{\{i \in X\}} \sum_{j=k_i}^{l_i-1} S_i^j(t) \ge \boldsymbol{\mathcal{Q}(t)}$.
Since for all $j < l_i - 1$, $S_i^j(t) \le S_i^{l_i-1}(t)$, thus
$\sum_{i=1}^{N} \mathbf{1}_{\{i \in X\}} (l_i - k_i) S_i^{l_i-1}(t) \ge \boldsymbol{\mathcal{Q}(t)}$.
Using Eq. (III.11), it follows that,

$$\sum_{i=1}^{N} \mathbf{1}_{\{i \in X\}} (F_X) S_i^{l_i-1}(t) \ge \boldsymbol{\mathcal{Q}(t)}. \quad \text{(III.19)}$$

and hence,

$$\sum_{i=1}^{N} \mathbf{1}_{\{i \in X\}} \sum_{j=0}^{l_i-1} Q_i^j \ge \frac{\boldsymbol{\mathcal{Q}(t)}}{F_X}. \quad \text{(III.20)}$$

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 $l_i \le j < |P_i|$,

$$\mathbf{E}[Q_i^j] \ge \lambda_i. \quad \text{(III.21)}$$

Taking expectations on both sides of Eq. (III.20) and using Eq. (III.21), we obtain,

$$\sum_{i=1}^{N} \mathbf{1}_{\{i \in X\}} \sum_{j=0}^{|P_i|} \mathbf{E}[Q_i^j] \ge \frac{\mathbf{E}[\mathcal{Q}(t)]}{F_X} + \sum_{i=1}^{N} \mathbf{1}_{\{i \in X\}} (\lambda_i(|P_i| - l_i). \quad \text{(III.22)}$$

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

$$\frac{\sum_{i=1}^{N} \mathbf{1}_{\{i \in X\}} \sum_{j=0}^{l_i-1} \mathbf{E}[Q_i^j]}{\lambda_X} \ge \frac{\mathbf{E}[\widetilde{D_X}]}{F_X} + \frac{\sum_{i=1}^{N} \mathbf{1}_{\{i \in X\}} \lambda_i(|P_i| - l_i)}{\lambda_X}. \quad \text{(III.23)}$$

■

C. *Analysis of the complete system*

We now present a greedy algorithm which computes a lower bound on the average delay for a system containing multiple bottlenecks. The (1, X)-bottlenecks correspond to cliques in the conflict graph [10]. 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 higher.

Algorithm 1 proceeds by greedily searching for a (K, X)-bottleneck that yields the maximum lower bound. The flows passing through the chosen bottleneck are removed and the process is repeated until all the flows are used. Thus it decomposes the wireless network into several single queue systems. The average delay of the system can then be easily computed. Note that the decomposition obtained by the greedy algorithm is not the optimal decomposition. The optimal decomposition can alternately be obtained by using a dynamic programming approach with the cost of increased computation complexity.

Algorithm 1 Computing the Lower Bound

```
Z ← {1, 2 ... N}
BOUND ← 0
repeat
    Find the (K, X)-bottleneck which maximizes E[D_X]
    BOUND ← BOUND + E[D_X]
    Z ← Z \ i : i ∈ X
until Z = Φ
return BOUND
```

The lower bound analysis may be loose on account of the following. Firstly, inequality (III.19) 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

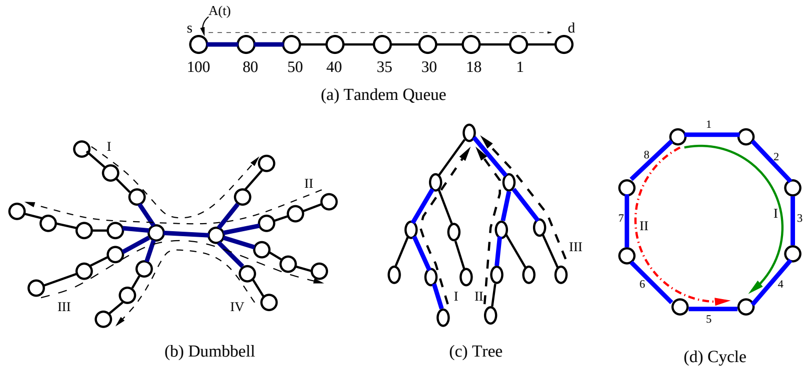

Fig. 4. Illustration of the Lower Bound analysis (bottlenecks used in the analysis have been highlighted).

through multiple bottlenecks. Thirdly, we assume that the queueing 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 delay.

IV. Illustrative Examples

We now demonstrate our methodology on a variety of examples shown in Fig. 4. 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 II-A) 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 [2], [21] in Sections IV-B and IV-C.

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 II-A. Cplex [8], an integer-programming solver, was used to compute the maximum weight matchings. We simulate the Tandem Queue in Section IV-A under a 1-hop interference model because a delay optimal policy [19] for this case is known. 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. 6 and 7 .

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 the Zipf law with power exponent 1.25 and support [1,2,3,...,100]. Heavy tailed distributions like Zipf, have been found to model the Internet traffic [6]. 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 [5] to obtain the expected delay for the single queue systems.

A. Tandem Queue

We consider a stream of packets flowing over the wireless links in tandem, as shown in Fig. 4(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 $\frac{\mathbf{E}[\widehat{D_X}]}{F_X}$ is the same for all exclusive sets in the system. The lower bound for the above arrival process is given by,

$$\mathbf{E}[D_t] \geq \frac{2(\zeta - 0.5)\lambda(1-\lambda)^2}{1-2\lambda} - \frac{\lambda^2(1-\lambda)}{1-2\lambda} + \zeta\lambda^2 - \lambda(\zeta - 1) + 6 \tag{IV.24}$$

where λ is the arrival rate in packets/slot and $\zeta = \frac{\mathbf{E}[a^2]}{2\mathbf{E}[a]} + 0.5$ is the mean residual time in an active period. Simulation results in Fig. 5 show that this lower bound virtually coincides with the delay performance of the optimal scheme [19].

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. 4(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. 5.

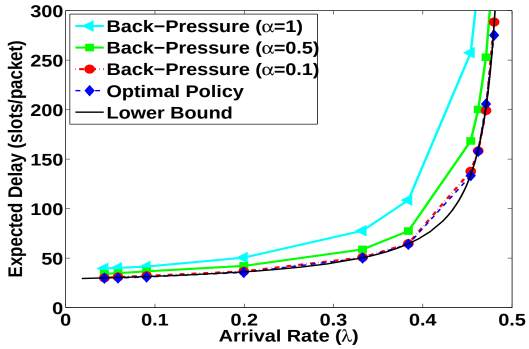

Fig. 5. Simulation results for Tandem Queue

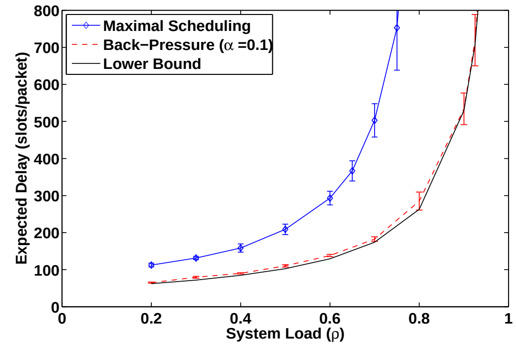

Fig. 6. Simulation results for Dumbbell Topology

Indeed, the scheme in [19] is valid only for the tandem queue under 1-hop interference model. It has been suggested in the literature [11], [18] 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. Dumbbell topology

Consider a dumbbell topology (Fig. 4(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. 4(b). The performance of back-pressure policy ($\alpha = 0.1$) is compared with that of the lower bound in Fig. 6. 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 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 [2], [21].

C. Tree Topology

Consider a tree topology with three flows converging at the root of the tree shown in Fig. 4(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.19) 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. 7, 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.

D. Cycle Topology

We now illustrate the application of the lower bound analysis to a (2, X)-bottleneck. For the given cycle network in Fig. 4(d), no more than two links can be scheduled at any time under 2-hop interference constraints, i.e. for $X = \{1, 2, 3 \ldots, 8\}$, $I_X(t) \leq 2$. It can be easily verified that the analysis presented in Section III-B can be used to reduce the 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. 8.

E. Analysis of the Example in Fig. 1

In this example, we analyze the wireless grid with randomly generated flows initially described in Fig. 1. 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 $\boldsymbol{\lambda}$. We find that depending on the input load vector, Algorithm 1 computes different decompositions 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:
$\boldsymbol{\lambda} = (0.12, 0.15, 0.12, 0.06, 0.12, 0.15, 0.12)$ packets/slot.
For the given load vector, Algorithm 1 computes the decomposition {{II, VI};{I, V};{III};{IV};{VII}}. Note that flow IV interferes with flows II and VI significantly, but

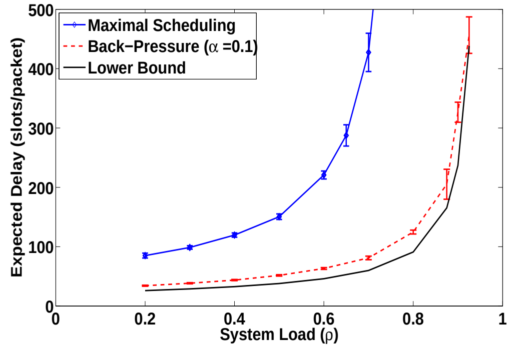

Fig. 7. Simulation results for Tree Topology

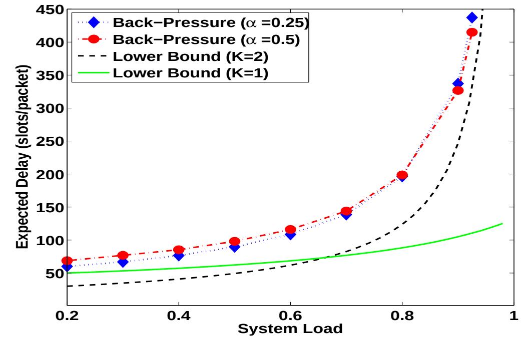

Fig. 8. Simulation results for Cycle Topology

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 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:
$\boldsymbol{\lambda} = (0.12, 0.15, 0.12, 0.0, 0.12, 0.15, 0.12)$ 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.

V. Discussion and Related Work

Much of the analysis [2], [7], [15] 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 the 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 stability of the system or to study the workload process in the heavy traffic regime. It has been shown in [4] 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 [7], [13], [16], [17] 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 [17] 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 systems have been obtained in [14], [20], [23] to calculate 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.

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. This technique captures the essential features of the wireless network and is useful since, in many cases, we can also find that the back-pressure policy performs close to the lower bound. Perhaps, the most important advantage of the lower bound is that it is simple to compute and can be used for analyzing a large class of arrival processes using known results in the queueing literature [5].

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 [10] 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 queue under the node exclusive interference model derived in [19], no result is known for other topologies and interference models.

In [22], 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, it is highly unlikely that the M/M/1 approximation could be accurate.

In [9], 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, $\boldsymbol{\lambda}$, 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 [11] 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 [18] 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 IV-D and IV-E, 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.

VI. Conclusion

In this paper, we have described a simple approach to reduce the bottlenecks in a multi-hop wireless to single queue systems to carry out the lower bound analysis. We emphasize that even in the wireline setting, obtaining results beyond the product form networks has been very difficult. 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 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. Although we do not explicitly input the information regarding the arrival processes or the bottlenecks to the back-pressure algorithm, the backlogs guide it appropriately to schedule resources carefully among the flows in the network. It is interesting to note that the back-pressure type of policies, which were designed primarily for achieving maximum throughput, can also be engineered to achieve good delay performance.

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] 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.

[3] J. G. Dai and W. Lin. Maximum pressure policies in stochastic processing networks. *Operations Research*, 53:197–218, 2005.

[4] J. G. Dai and W. Lin. Asymptotic optimality of maximum pressure policies in stochastic processing networks. Preprint, October 2007.

[5] H. Dupuis and B. Hajek. A simple formula for mean multiplexing delay for indep endent regenerative sources. *Queueing Systems Theory and Applications*, 16:195–239, 1994.

[6] 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.

[7] 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.

[8] I. Ilog. Solver cplex, 2007. http://www.ilog.com/products/cplex/.

[9] S. Jagabathula and D. Shah. Optimal delay scheduling in networks with arbitrary constraints. In *ACM SIGMETRIC/Performance*, June 2008.

[10] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. In *MOBICOM*, 2003.

[11] 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.

[12] T. Leandros 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.

[13] 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), Feburary 2001.

[14] X. Lin. On characterizing the delay performance of wireless scheduling algorithms. In *44th Annual Allerton Conference on Communication, Control and Computing*, September 2006.

[15] 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.

[16] 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.

[17] M. J. Neely. Delay analysis for maximal scheduling in wireless networks with bursty traffic. IEEE INFOCOM, 2008.

[18] D. Shah and D. Wischik. Heavy traffic analysis of optimal scheduling algorithms for switched networks. Preprint, 2008.

[19] L. Tassiulas and A. Ephremides. Dynamic scheduling for minimum delay in tandem and parallel constrained queueing models. *nnals of Operation Research, Vol. 48, 333-355*, 1993.

[20] 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.

[21] 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.

[22] Y. Xi and E. M. Yeh. Optimal capacity allocation, routing, and congestion control in wireless networks. In *International Symposium on Information Theory*, July 2006.

[23] 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.