

Delay-Based Network Utility Maximization

Michael J. Neely

University of Southern California

<http://www-rcf.usc.edu/~mjneely>

Abstract—It is well known that max-weight policies based on a queue backlog index can be used to stabilize stochastic networks, and that similar stability results hold if a delay index is used. Using Lyapunov Optimization, we extend this analysis to design a utility maximizing algorithm that uses explicit delay information from the head-of-line packet at each user. The resulting policy is shown to ensure deterministic worst-case delay guarantees, and to yield a throughput-utility that differs from the optimally fair value by an amount that is inversely proportional to the delay guarantee. Our results hold for a general class of 1-hop networks, including packet switches and multi-user wireless systems with time varying reliability.

I. INTRODUCTION

We consider the problem of scheduling for maximum throughput-utility in a 1-hop network with random packet arrivals and time varying channel reliability. Every slot, the network controller assesses the condition of its channels and selects a set of links for transmission. The success of each transmission depends on the collection of links selected and their corresponding reliabilities. The goal is to maximize a concave, non-decreasing function of the time average throughput on each link. Such a function represents a *utility function* that acts as a measure of fairness for the achieved throughput vector.

In the case when the traffic is inside the *network capacity region*, the utility-optimal throughput vector is simply the vector of arrival rates, and the problem reduces to a *network stability problem*. In this case, it is well known that the network can be stabilized by *max weight* policies that schedule links every slot to maximize a weighted sum of transmission rates, where the weights are queue backlogs. This is typically shown via a Lyapunov drift argument (see [1] and references therein). This technique for stable control of a queueing network was first used for link and server scheduling in [2][3], and has since become a powerful method to treat stability in different contexts, including switches and computer networks [4][5][6], wireless systems and ad-hoc mobile networks with rate and power allocation [7][8][9], and systems with probabilistic channel errors [10][11].

In the case when traffic is either inside or outside of the capacity region, it is known that the max-weight policy can be combined with a flow control policy to jointly stabilize the network while maximizing throughput-utility. This is shown in [1][12][13] via a Lyapunov Optimization argument, and in [14] via a fluid limit analysis. Utility optimization for the special case of “infinitely backlogged” sources is shown in

[15][16][17], and was perhaps first addressed for time-varying wireless downlinks without explicit queueing in [18][19].

The stability works [2]-[11] all use backlog-based transmission rules, as do the works in [1], [12]-[17] which treat joint stability and utility optimization. However, work in [20] introduces an interesting *delay-based* Lyapunov function for proving stability, where the delay of the head-of-line packet is used as a weight in the max-weight decision. This approach intuitively provides tighter control of the actual queueing delays. Indeed, a single head-of-line packet is scheduled based on the delay it has experienced, rather than on the amount of additional packets that arrived after it. This delay-based approach to queue stability is extended in [21], where the *Modified Largest Weighted Delay First* algorithm is developed, and in [22] which uses a delay-based exponential rule. However, [20]-[22] use delay-based rules only in the context of queue stability. To our knowledge, there are no prior works that use delay-based scheduling to address the important issue of joint stability and utility optimization.

This paper fills that gap. We use a delay-based Lyapunov function, and extend the analysis to treat joint stability and performance optimization via the Lyapunov Optimization technique from our prior work [1][12][13]. The extension is not obvious. Indeed, the flow control decisions in the prior work [1][12][13] are made immediately when a new packet arrives, which directly affects the drift of backlog-based Lyapunov functions. However, such decisions *do not* directly affect the delay value of the head-of-line packets, and hence do not directly affect the drift of delay-based Lyapunov functions. We overcome this challenge with a novel flow control policy that queues *all* arriving data, but makes packet dropping decisions just before advancing a new packet to the head-of-line. This policy is structurally different from the utility optimization works [1], [12]-[19]. We show that this new structure leads to deterministic guarantees on the worst-case delay of any non-dropped packet, while also yielding throughput-utility that is arbitrarily close to optimal. Specifically, for any integer $D \geq 2$, we can construct an algorithm that ensures all non-dropped packets have delay less than or equal to D slots, with total throughput-utility that differs from optimal by $O(1/D)$. The deterministic delay guarantee is particularly challenging to establish, and for this we introduce a new technique of *concavely extending a utility function*.

Similar $[O(1/D), O(D)]$ performance tradeoffs are shown for queue-based Lyapunov functions in our previous work [1][12][13] (see also [23][24] for deterministic queue backlog bounds, and [25][26] for improved tradeoffs), but these guar-

This material is supported in part by one or more of the following: the DARPA IT-MANET program grant W911NF-07-0028, the NSF Career grant CCF-0747525.

antees apply only to queue size, rather than delay.¹ The deterministic delay guarantees we obtain in this present paper are quite strong and show the advantages of our new flow control structure. However, a disadvantage is that admit/drop decisions are delayed until a packet is at the head-of-line, rather than being determined immediately upon arrival. Further, due to correlation issues unique to this delay-based scenario, analysis is simplified if we assume the scheduler knows the vector of arrival rates to each link (although we also treat the case when arrival rates are unknown). Nevertheless, it is important to analyze these delay-based policies because they improve our understanding of network delay, and because the deterministic guarantees they offer are useful for many practical systems.

While our algorithm can be used to enforce any desired delay guarantee, it is important to emphasize that it does *not* maximize throughput-utility subject to this guarantee. Such a problem can be addressed with dynamic programming and Markov decision theory, which bring with them the *curse of dimensionality* (see structural results and approximations in [28] and weighted stochastic shortest path approaches in [29]). In the present paper, we claim only that the achieved utility is within $O(1/D)$ of the largest possible utility achievable by any stabilizing algorithm. However, because (for large D) our utility is close to this ideal utility value, it is even closer to the maximum utility that can be achieved subject to the worst-case delay constraint. Further, our approach offers the low complexity advantages associated with Lyapunov drift and Lyapunov Optimization. Specifically, the policy makes real-time transmission decisions based only on the current system state, and does not require a-priori knowledge of the channel state probabilities. The flow control decisions here can also be implemented in a distributed fashion at each link, as is the case with most other Lyapunov based utility optimization algorithms (this is not necessarily the case for dynamic programming or Markov decision theory approaches).

It is important to distinguish our work, which considers *actual network delay*, with work that approximates network delay as a convex function of a flow rate (such as in [30][27]). While it is known that average queue congestion and delay is convex if traffic is probabilistically split [31], this is not necessarily true (or relevant) for dynamically controlled networks, particularly when the control depends on the queue backlogs and delays themselves. Actual network delay problems involve not only optimization of rate based utility functions, but engineering of the Lagrange multipliers (which are related to queue backlogs) associated with those utility functions.

In the next section we present the network model. Section III develops our main algorithm for delay-based flow control and utility optimization. The utility performance and worst-case delay guarantees are proven in Section IV.

II. NETWORK MODEL

The network is assumed to be a 1-hop network that operates in discrete time with normalized timeslots $t \in \{0, 1, 2, \dots\}$. There are L links, and packets arrive randomly every slot and

are queued separately for transmission over each link. We let $\mathbf{A}(t) = (A_1(t), \dots, A_L(t))$ be the process of random packet arrivals, where $A_l(t)$ is the number of packets that arrive to link l on slot t . For simplicity, we assume all packets have fixed size, and that there is at most one packet arrival to each link per slot, so that $A_l(t) \in \{0, 1\}$ for all links l and slots t . The arrival vector $\mathbf{A}(t)$ is assumed to be i.i.d. over slots, and further the arrival processes $A_l(t)$ for different links in each slot are assumed to be independent. Let $\mathbf{Q}(t) = (Q_1(t), \dots, Q_L(t))$ denote the integer number of packets currently stored in each of the L queues. All packets are marked with their integer arrival slot, which is used to determine their delay in the system. The one-step queueing equation for each link l is:

$$Q_l(t+1) = \max[Q_l(t) - \mu_l(t) - D_l(t), 0] + A_l(t) \quad (1)$$

where $\mu_l(t)$ represents the amount of packets successfully served on slot t , and $D_l(t)$ represents the number of packets *dropped* on slot t . A packet can be dropped at any time, although in our specific algorithm we impose a 2-stage structure that first makes a transmission decision and then makes a dropping decision in reaction to the feedback obtained from the transmission.

A. Time Varying Link Reliability

For simplicity, we assume that each link can transmit at most one packet per slot, so that $\mu_l(t) \in \{0, 1\}$ for all links l and all slots t . Let $\mathbf{x}(t) = (x_1(t), \dots, x_L(t))$ denote a *transmission vector*, where $x_l(t) \in \{0, 1\}$, and $x_l(t) = 1$ if link l attempts transmission on slot t . Let \mathcal{X} denote the set of all allowable link transmission vectors, possibly being the set of all 2^L such vectors, but also possibly incorporating some constraints (such as permutation constraints for $N \times N$ packet switches). In principle, it is useful to assume a link can transmit even if it does not have a packet, in which case a *null packet* is transmitted. Let $\mathbf{S}(t) = (S_1(t), \dots, S_L(t))$ denote a link condition vector for slot t , which determines the probability of successful transmission on each slot. Specifically, given particular $\mathbf{x}(t)$ and $\mathbf{S}(t)$ vectors, the probability of successful transmission on link l is given by a *reliability function*:

$$Pr[\text{link } l \text{ success} | \mathbf{x}(t), \mathbf{S}(t)] = \Psi_l(\mathbf{x}(t), \mathbf{S}(t)) \quad (2)$$

The reliability function $\Psi_l(\mathbf{x}, \mathbf{S})$ for each $l \in \{1, \dots, L\}$ is general and is assumed only to take real values between 0 and 1 (representing probabilities), and to have the property that $\Psi_l(\mathbf{x}, \mathbf{S}) = 0$ whenever $x_l = 0$. We assume that the channel condition vector $\mathbf{S}(t)$ is i.i.d. over slots, taking values in a set \mathcal{S} of arbitrary cardinality, and that $\mathbf{S}(t)$ is known to the network controller at the beginning of each slot t . In practice, $\mathbf{S}(t)$ represents the result of a channel measurement or estimation that is done every slot. The estimate might be inexact, in which case the reliability function $\Psi_l(\mathbf{x}(t), \mathbf{S}(t))$ represents the probability that the actual network channels on slot t are sufficient to support the attempted transmission over link l (given $\mathbf{x}(t)$ and the estimate $\mathbf{S}(t)$ for slot t).

We assume the reliability function is known. Recent online techniques for estimation of packet error rates are considered in [32]. In the context of [32], a number of other decision

¹Of course *average* delay and *average* backlog are directly related through Little's Theorem [27], but this is not true for worst-case backlog and delay.

parameters to be chosen on each slot also affect reliability, such as modulation, power levels, subband selection, coding type, etc. These choices can be represented as a parameter space \mathcal{I} . In this case, the reliability function can be extended to include the parameter choice $I(t) \in \mathcal{I}$ made every slot: $\Psi_l(\mathbf{x}(t), \mathbf{S}(t), I(t))$. This does not change our mathematical analysis (see also Remark 1 in Section III-F), although for simplicity we focus on the reliability function structure of (2).

We assume that ACK/NACK information is given at the end of the slot to inform each link if its transmission was successful or not. Packets that are not successful do not leave the queue (unless they are dropped in a packet drop decision). With this model of link success, the transmission variable $\mu_l(t)$ in (1) is given by:

$$\mu_l(t) = x_l(t)1_l(t)$$

where $1_l(t)$ is an indicator variable that is 1 if the transmission over link l is successful, and 0 otherwise. That is:

$$1_l(t) = \begin{cases} 1 & \text{with probability } \Psi_l(\mathbf{x}(t), \mathbf{S}(t)) \\ 0 & \text{with probability } 1 - \Psi_l(\mathbf{x}(t), \mathbf{S}(t)) \end{cases}$$

The successes/failures over each link on slot t are assumed to be independent of past history given the current $\mathbf{x}(t)$ and $\mathbf{S}(t)$ values. The successes/failures might be correlated over each link. This is not captured in the $\Psi_l(\mathbf{x}, \mathbf{S})$ functions alone, and can only be fully described by a joint success distribution function for all 2^L possible success/failure outcomes for a given \mathbf{x} and \mathbf{S} . However, it turns out that the network capacity region, and hence the associated maximum utility point, is independent of such inter-link success correlations [11]. Hence, it suffices to use only the *marginal* distribution functions $\Psi_l(\mathbf{x}, \mathbf{S})$ for each $l \in \{1, \dots, L\}$.

B. Examples of Packet Switches and Wireless Networks

The above model applies to a wide class of 1-hop networks. For example, it applies to the $N \times N$ packet switch models of [4][6] by defining $\mathbf{S}(t)$ to be a null vector (so that there is no notion of channel variation), and by defining the set \mathcal{X} of all allowable link vectors to be the set of all vectors that satisfy the permutation constraints associated with the $N \times N$ crossbar. For wireless networks with interference but without time varying channels, the set \mathcal{X} can be defined as all link activations that do not interfere with each other (i.e., that do not produce *collisions*), as in [2]. The reliability function $\Psi_l(\cdot)$ can be used to extend the model to treat cases where interfering links result in probabilistic reception (rather than collision).

Further, the opportunistic scheduling systems of [3] with time-varying ON/OFF channels can be modeled with $\mathbf{S}(t)$ being the vector of ON/OFF channel states on each slot, and with the function $\Psi_l(\mathbf{x}, \mathbf{S})$ taking the value 1 whenever $x_l = 1$ and $S_l = ON$, and 0 otherwise. Finally, the model supports probabilistic reception in the case when the link reliability can vary from slot to slot.

A simple example is when $S_l(t)$ represents the current probability that a link l transmission would be successful, so that:

$$\Psi_l(\mathbf{x}(t), \mathbf{S}(t)) = \begin{cases} S_l(t) & \text{if } x_l(t) = 1 \\ 0 & \text{if } x_l(t) = 0 \end{cases}$$

This example has the success probability over link l a pure function of $x_l(t)$ and $S_l(t)$, and hence implicitly assumes that the set \mathcal{X} limits all simultaneous link transmissions to orthogonal channels. More complex inter-channel interference models can be described by more complex $\Psi_l(\mathbf{x}, \mathbf{S})$ functions.

III. DELAY-BASED FLOW CONTROL

Let $\boldsymbol{\lambda} = \mathbb{E}\{\mathbf{A}(t)\}$ be the vector of arrival rates, so that $\lambda_l = \mathbb{E}\{A_l(t)\}$ is the arrival rate to link l (in units of packets/slot). The *network capacity region* Λ is defined as the closure of the set of all long-term throughput vectors that the system can support. The set Λ is known to be the same as the closure of the set of all arrival rate vectors $\boldsymbol{\lambda}$ for which there exists a stabilizing scheduling algorithm, subject to the constraint that *the flow controllers are turned off* (so that no packets are dropped and $D_l(t) = 0$ for all l and all t) [1] [11]. Specifically, in [11] it is shown that the set Λ is given by the set of all time average transmission rates that can be achieved by stationary and randomized algorithms, called *S-only* algorithms, that observe $\mathbf{S}(t)$ every slot t and choose a (possibly random) transmission vector $\mathbf{x}(t) \in \mathcal{X}$ according to a probability distribution that depends only on the observed channel state $\mathbf{S}(t)$. Thus, for every vector $\mathbf{r} \in \Lambda$, with $\mathbf{r} = (r_1, \dots, r_L)$, there is a *S-only* policy $\mathbf{x}^*(t)$, with a corresponding random service vector $\boldsymbol{\mu}^*(t) = (x_1^*(t)1_1^*(t), \dots, x_L^*(t)1_L^*(t))$ that yields for each $l \in \{1, \dots, L\}$:

$$r_l = \mathbb{E}\{\mu_l^*(t)\} = \mathbb{E}\{x_l^*(t)\Psi_l(\mathbf{x}^*(t), \mathbf{S}(t))\} \quad (3)$$

where the expectation in (3) is with respect to the distribution of $\mathbf{S}(t)$ and the distribution of $\mathbf{x}^*(t)$ given $\mathbf{S}(t)$.

A. The Optimization Objective

Let $g(\mathbf{y})$ be a continuous and concave utility function of the L -dimensional vector $\mathbf{y} = (y_1, \dots, y_L)$, where \mathbf{y} is used to represent the time average throughput on each link (in units of packets/slot). The function can take positive or negative values, and is assumed to be defined over the hyper-cube $\mathbf{0} \leq \mathbf{y} \leq \mathbf{1}$, where inequality is taken entrywise, and $\mathbf{0}$ and $\mathbf{1}$ are vectors with all entries equal to 0 and 1, respectively. Assume that $g(\mathbf{y})$ is non-decreasing in each entry y_l . An example is the *separable* utility function:

$$g(\mathbf{y}) = \sum_{l=1}^L g_l(y_l) \quad (4)$$

where for each link l , $g_l(y_l)$ is a concave and non-decreasing function of y_l , defined over the interval $0 \leq y_l \leq 1$. We make the following additional assumption.

Assumption 1: For each $l \in \{1, \dots, L\}$, the l th right partial derivative of $g(\mathbf{y})$, over all $\mathbf{y} \in [0, 1]^L$ such that $y_l < 1$, is bounded above by a finite constant ν_l , where $\nu_l \geq 0$.²

Assumption 1 implies that for any vectors \mathbf{y} and \mathbf{w} such that $\mathbf{y} \in [0, 1]^L$, $\mathbf{w} \in [0, 1]^L$, and $\mathbf{y} + \mathbf{w} \in [0, 1]^L$ we have:

$$g(\mathbf{y} + \mathbf{w}) \leq g(\mathbf{y}) + \sum_{l=1}^L \nu_l w_l \quad (5)$$

²Right partial derivatives exist for any concave function, including non-differentiable functions such as $g(\mathbf{y}) = \min\{y_1, \dots, y_L\}$.

Note that Assumption 1 does *not* hold for the logarithmic utility function $\sum_l \log(y_l)$ associated with proportional fairness [33]. However, it *does* hold for the following useful utility function example, which is often a good alternative way to treat network fairness:

$$g(\mathbf{y}) = \sum_l \nu_l \log(1 + y_l) \quad (6)$$

Further, it holds for *approximations* of the proportional fairness function, such as using $\sum_l \log(\epsilon + y_l)$ for some small value of ϵ .

For each link l , define $Y_l(t)$ as:

$$Y_l(t) = \lambda_l - D_l(t) \quad (7)$$

Let $\bar{\mathbf{y}}(t)$ be the time average expectation of $Y_l(t)$ over t slots:

$$\bar{\mathbf{y}}(t) \triangleq \lambda_l - \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E} \{ \mathbf{D}(\tau) \} \quad (8)$$

where $\mathbf{D}(\tau) = (D_1(\tau), \dots, D_L(\tau))$ is the drop vector for slot τ . Let $\bar{\mathbf{y}}$ denote the limit of $\bar{\mathbf{y}}(t)$ as $t \rightarrow \infty$ (assumed temporarily to exist). The vector $\bar{\mathbf{y}}$ is the difference between the rate of arrivals and packet drops, and hence (if queues are stable) represents the throughput vector. The goal is to design a delay-based transmission scheme with packet dropping that solves the following problem:

$$\text{Maximize:} \quad g(\bar{\mathbf{y}}) \quad (9)$$

$$\text{Subject to:} \quad \bar{\mathbf{y}} \in \Lambda \quad (10)$$

$$0 \leq \bar{y}_l \leq \lambda_l \quad \text{for all } l \in \{1, \dots, L\} \quad (11)$$

Let g^* be the supremum utility value for the above problem. In addition to striving to achieve a utility that is close to g^* , we desire the actual delays of non-dropped packets to be deterministically bounded.

B. The Concavely Extended Utility Function

Suppose that $g(\mathbf{y})$ satisfies Assumption 1, and define the *concave extension* of $g(\mathbf{y})$ as the function $\hat{g}(\mathbf{y})$ defined over the extended hyper-cube $-1 \leq \mathbf{y} \leq \mathbf{1}$ as follows:

$$\hat{g}(\mathbf{y}) \triangleq g(\max[\mathbf{y}, \mathbf{0}]) + \sum_{l=1}^L \nu_l \min[y_l, 0]$$

where $\max[\mathbf{y}, \mathbf{0}]$ represents an entry-wise max:

$$\max[\mathbf{y}, \mathbf{0}] \triangleq (\max[y_1, 0], \max[y_2, 0], \dots, \max[y_L, 0])$$

It can be shown that $\hat{g}(\mathbf{y})$ is concave and non-decreasing in each entry over the extended hyper-cube $-1 \leq \mathbf{y} \leq \mathbf{1}$, and that:

$$g(\mathbf{y}) = \hat{g}(\mathbf{y}) \quad \text{whenever } \mathbf{0} \leq \mathbf{y} \leq \mathbf{1}$$

Further, because (5) holds, it can be shown that for any vector \mathbf{y} such that $-1 \leq \mathbf{y} \leq \mathbf{1}$:

$$\hat{g}(\mathbf{y}) \leq \hat{g}(\mathbf{y}') + \nu_l (y_l + 1) \quad (12)$$

where the vector \mathbf{y}' is formed from the vector \mathbf{y} by replacing the single entry y_l with -1 .

In the case when $g(\mathbf{y})$ has the separable form (4), the concave extension is given by $\hat{g}(\mathbf{y}) = \sum_l \hat{g}_l(y_l)$, where each function $\hat{g}_l(y_l)$ concavely extends the function $g_l(y_l)$, originally defined over the interval $0 \leq y_l \leq 1$, to the interval $-1 \leq y_l \leq 1$, as shown in Fig. 1. This method of concavely

extending the utility function is crucial to engineer the network delays to be bounded (in particular, it is needed to allow $\gamma_l(t) = -1$ in (24)).

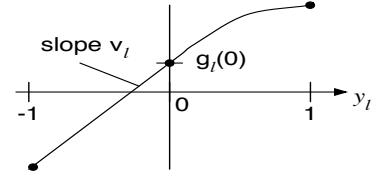


Fig. 1. An illustration of the concave extension of $g_l(y_l)$.

C. Problem Transformation with Virtual Queues

It is not difficult to show that the stochastic network optimization problem (9)-(11) can be transformed using a vector $\boldsymbol{\gamma}(t)$ of *auxiliary variables* [1] [12] that are chosen every slot according to the constraints $-1 \leq \gamma_l(t) \leq 1$. The transformed problem is:

$$\text{Maximize:} \quad \hat{g}(\bar{\boldsymbol{\gamma}}) \quad (13)$$

$$\text{Subject to:} \quad \bar{Q}_l < \infty \quad \text{for all } l \in \{1, \dots, L\} \quad (14)$$

$$\bar{y}_l \geq \bar{\gamma}_l \quad \text{for all } l \in \{1, \dots, L\} \quad (15)$$

$$-1 \leq \bar{\gamma}_l \leq 1 \quad \text{for all } l \in \{1, \dots, L\} \quad (16)$$

$$\bar{\mathbf{Q}} \text{ and } \bar{\mathbf{y}} \text{ are achievable on the network} \quad (17)$$

where \bar{Q}_l is defined:

$$\bar{Q}_l \triangleq \limsup_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E} \{ Q_l(\tau) \}$$

We say that a non-negative discrete time stochastic process $Q_l(t)$ is *strongly stable* if $\bar{Q}_l < \infty$.

This transformation can be intuitively understood as follows: The constraint (11) automatically holds for any achievable control policy, as the throughput cannot be larger than the raw arrival rate, and hence is satisfied whenever (17) holds. The constraint (10) is ensured by the stability constraint (14) in the transformed problem. Finally, one can always choose the auxiliary vector $\boldsymbol{\gamma}(t) = \mathbf{y}(t)$ to ensure that (15) and (16) are satisfied (note that $0 \leq \bar{y}_l \leq 1$ for all l , because arrival rates cannot be larger than 1). The fact that $g(\mathbf{y})$ is non-decreasing in each entry and that $\hat{g}(\mathbf{y}) = g(\mathbf{y})$ whenever $\mathbf{0} \leq \mathbf{y} \leq \mathbf{1}$ ensures that it suffices to consider all constraints (15) holding with equality, so that any control algorithm that solves (13)-(17) also solves (9)-(11).

The auxiliary variables $\boldsymbol{\gamma}(t)$ are important for solving problems of maximizing a concave function of a time average, and are crucial for network utility maximization with randomly arriving traffic [1] [12].³ To ensure that the constraints (15) are satisfied, we use a *virtual queue* $Z_l(t)$ for each link l , with update equation as follows:

$$Z_l(t+1) = \max[Z_l(t) - \lambda_l + D_l(t) + \gamma_l(t), 0] \quad (18)$$

Stabilizing this virtual queue ensures that the time average value of $Y_l(t)$, defined in (7), is greater than or equal to the

³Cases of “infinitely backlogged sources” do not need such auxiliary variables [13] [12].

time average of $\gamma_l(t)$ [24], which ensures (15). Specifically, using the definition $Y_l(t)$ in (7), from (18) it is clear that:

$$Z_l(t+1) \geq Z_l(t) - Y_l(t) + \gamma_l(t)$$

and hence (by summing the above over $\tau \in \{0, \dots, t-1\}$ and dividing by t):

$$\frac{Z_l(t) - Z_l(0)}{t} + \frac{1}{t} \sum_{\tau=0}^{t-1} Y_l(\tau) \geq \frac{1}{t} \sum_{\tau=0}^{t-1} \gamma_l(\tau)$$

Taking expectations of both sides and using $Z_l(0) = 0$ yields:

$$\frac{\mathbb{E}\{Z_l(t)\}}{t} + \bar{y}_l(t) \geq \bar{\gamma}_l(t) \quad (19)$$

where $\bar{y}_l(t)$ is the time average expected value of $Y_l(\tau)$, defined in (8), and $\bar{\gamma}_l(t)$ is defined similarly. It follows from (19) that if $\mathbb{E}\{Z_l(t)\}/t \rightarrow 0$ (a property that is satisfied if $Z_l(t)$ is strongly stable, as shown in [24]), and if $\bar{y}_l(t)$ and $\bar{\gamma}_l(t)$ have well defined limits \bar{y}_l and $\bar{\gamma}_l$, then $\bar{y}_l \geq \bar{\gamma}_l$, so that the constraints (15) are satisfied.

Implementation of the virtual queue (18) assumes the arrival rates λ_l are known for each link l . If these are unknown, one can modify the virtual queue update rule (18) to:

$$Z_l(t+1) = \max[Z_l(t) - A_l(t-W) + D_l(t) + \gamma_l(t), 0]$$

where W is a suitably large positive integer (using $W = \lceil V \max_l \{\nu_l\} \rceil + 2$ works well). The use of a W -shifted arrival process $A_l(t-W)$ is unique to this delay-based analysis, and is required to handle subtle correlation issues between packet inter-arrival times and virtual queue states. For simplicity, here we analyze only the case when update equation (18) is used.

D. The Delay-Based Lyapunov Function

We now impose the following structure on our control policy: Every slot, a packet transmission decision is made *first*. If a transmission over link l is successful (so that $\mu_l(t) = 1$), then the packet is removed from the queue and no packet is dropped from link l (so that $D_l(t) = 0$). Else, if link l either did not attempt transmission or if its transmission was unsuccessful, we can possibly decide to drop the packet, but no other packet can be dropped from link l . Thus, every slot t we have $0 \leq \mu_l(t) + D_l(t) \leq 1$. We show later that this structure does not hinder our maximum utility objective. Further, it is useful to consider the possibility of transmitting or dropping a *null packet* when the queue is empty, so that $\mu_l(t)$ and $D_l(t)$ in principle can be chosen independently of queue backlog.

Let $H_l(t)$ represent the waiting time of the head-of-line packet in link l on slot t , and define $H_l(t) = 0$ if there are no packets in link l at this time. A new packet that arrives to an empty queue on slot t is not placed to the head-of-line until the next slot, and is designated to have a waiting time of 1 at slot $t+1$. Define $\alpha_l(t)$ as an indicator variable that is 1 if $Q_l(t) > 0$, and is zero if the queue is empty. Let $\beta_l(t) = 1 - \alpha_l(t)$. Similar to [20], we observe that $H_l(t)$ satisfies the following update rule:

$$H_l(t+1) = \alpha_l(t) \max[H_l(t) + 1 - (\mu_l(t) + D_l(t))T_l(t), 0] + \beta_l(t)A_l(t) \quad (20)$$

where $T_l(t)$ represents the *interarrival time* between the head-of-line packet and the subsequent packet (possibly unknown to the network controller if the subsequent packet has not yet arrived). Because arrivals are Bernoulli, $T_l(t)$ is a geometric random variable with success probability λ_l , that takes values in the set $\{1, 2, 3, \dots\}$.

The equation (20) can be understood as follows: If $\alpha_l(t) = 0$, then $\beta_l(t) = 1$ so that queue l is empty. In this case, the value of $H_l(t+1)$ is 1 if and only if there is a new arrival on slot t . Alternatively, if $\alpha_l(t) = 1$, then $\beta_l(t) = 0$. Suppose in this case that the head-of-line packet is neither served nor dropped (so that $\mu_l(t) + D_l(t) = 0$). Then its delay increases by 1, as described by (20). On the other hand, if the head-of-line packet is either served or dropped (so that $\mu_l(t) + D_l(t) = 1$), then the *next* packet enters the head-of-line, with a total waiting time equal to $H_l(t) + 1 - T_l(t)$ (where the additional “+1” comes because this operation takes one more slot). The above dynamics also capture the possibility that the inter-arrival time is greater than or equal to $H_l(t) + 1$, in which case the queue is empty on slot $t+1$ with $H_l(t+1) = 0$.

Without loss of generality, assume that $\lambda_l > 0$ for all links $l \in \{1, \dots, L\}$ (else, just remove the links that have no traffic). Define $\Theta(t) \triangleq [\mathbf{Z}(t); \mathbf{H}(t)]$, where $\mathbf{Z}(t)$ and $\mathbf{H}(t)$ are vectors of the virtual queues in (18) and the head-of-line values in (20). We use the following non-negative Lyapunov function:

$$L(\Theta(t)) \triangleq \frac{1}{2} \sum_{l=1}^L Z_l(t)^2 + \frac{1}{2} \sum_{l=1}^L \lambda_l H_l(t)^2$$

E. Minimizing the Drift-Minus-Utility

Define $\Delta(\Theta(t))$ as the *one-step conditional Lyapunov drift*:

$$\Delta(\Theta(t)) \triangleq \mathbb{E}\{L(\Theta(t+1)) - L(\Theta(t)) \mid \Theta(t)\}$$

Using our Lyapunov Optimization framework in [1], our strategy is to make transmission and packet dropping decisions to minimize a bound on the following drift-minus-utility expression every slot:

$$\Delta(\Theta(t)) - V \mathbb{E}\{\hat{g}(\gamma(t)) \mid \Theta(t)\}$$

where V is a non-negative control parameter that is chosen as desired, and will affect an explicit utility-delay tradeoff. We have the following preliminary lemmas:

Lemma 1: Every slot t , for any value of $\Theta(t)$, and under any control policy, the Lyapunov drift satisfies:

$$\begin{aligned} \Delta(\Theta(t)) &\leq B - \sum_l Z_l(t) \mathbb{E}\{Y_l(t) - \gamma_l(t) \mid \Theta(t)\} \\ &\quad - \sum_l \lambda_l H_l(t) \mathbb{E}\{(\mu_l(t) + D_l(t))T_l(t) - 1 \mid \Theta(t)\} \end{aligned}$$

where B is a finite constant.

Proof: The proof follows by squaring (18) and (20) and using $\alpha_l(t)H_l(t) = H_l(t)$, $\alpha_l(t)\beta_l(t) = 0$, $\mu_l(t)^2 = \mu_l(t)$, $D_l(t)^2 = D_l(t)$, and is omitted for brevity. ■

Lemma 2: Every slot t , for any value of $\Theta(t)$, and under any control policy for which $T_l(t)$ is independent of $D_l(t)$,

$\mu_l(t)$, and $\Theta(t)$ for all links l , we have:

$$\begin{aligned} \Delta(\Theta(t)) - V\mathbb{E}\{\hat{g}(\gamma(t)) \mid \Theta(t)\} &\leq B - V\mathbb{E}\{\hat{g}(\gamma(t)) \mid \Theta(t)\} \\ &\quad - \sum_l Z_l(t)\mathbb{E}\{\lambda_l - D_l(t) - \gamma_l(t) \mid \Theta(t)\} \\ &\quad - \sum_l H_l(t)\mathbb{E}\{\mu_l(t) + D_l(t) - \lambda_l \mid \Theta(t)\} \end{aligned}$$

where B is the same constant from Lemma 1.

Proof: Let $\chi(t)$ represent $[\Theta(t); (\mu_l(t) + D_l(t))]$. Using the law of iterated expectations, we have:

$$\begin{aligned} &\mathbb{E}\{(\mu_l(t) + D_l(t))T_l(t) \mid \Theta(t)\} \\ &= \mathbb{E}\{\mathbb{E}\{(\mu_l(t) + D_l(t))T_l(t) \mid \chi(t)\} \mid \Theta(t)\} \\ &= \mathbb{E}\{(\mu_l(t) + D_l(t))\mathbb{E}\{T_l(t) \mid \chi(t)\} \mid \Theta(t)\} \\ &= \frac{1}{\lambda_l}\mathbb{E}\{(\mu_l(t) + D_l(t)) \mid \Theta(t)\} \end{aligned}$$

where we have used the fact that $T_l(t)$ is independent of $\chi(t)$ and is a geometric random variable with $\mathbb{E}\{T_l(t)\} = 1/\lambda_l$. Lemma 2 follows by plugging this identity into Lemma 1 and subtracting $V\mathbb{E}\{\hat{g}(\gamma(t)) \mid \Theta(t)\}$ from both sides. ■

Our dynamic policy below makes control decisions for $\gamma(t)$, $D(t)$, and $\mathbf{x}(t)$ (and hence $\mu(t)$) to minimize the right hand side of the drift-minus-utility bound in Lemma 2.

F. The Delay-Based Flow Control and Scheduling Algorithm

Every slot t , observe $\mathbf{Z}(t)$, $\mathbf{H}(t)$, and $\mathbf{S}(t)$, and perform the following operations, described as four control phases:

- 1) *Auxiliary Variable Selection:* Choose $\gamma(t) = (\gamma_1(t), \dots, \gamma_L(t))$ as the solution to the following:

$$\begin{aligned} &\text{Maximize:} && V\hat{g}(\gamma(t)) - \sum_l Z_l(t)\gamma_l(t) \\ &\text{Subject to:} && -1 \leq \gamma_l(t) \leq 1 \quad \text{for all } l \in \{1, \dots, L\} \end{aligned}$$

In the case of the separable utility function (4), this amounts to solving L single-variable concave optimizations over an interval, and has a closed form solution when the derivative of $\hat{g}_l(\gamma_l)$ has a closed form inverse.

- 2) *Transmission Scheduling:* Observe $\Theta(t)$, $\mathbf{S}(t)$ and choose a transmission vector $\mathbf{x}(t)$ to solve the following:

$$\begin{aligned} &\text{Maximize:} && \sum_l x_l(t) \min[H_l(t), Z_l(t)]\Psi_l(\mathbf{x}(t), \mathbf{S}(t)) \\ &\text{Subject to:} && \mathbf{x}(t) \in \mathcal{X} \end{aligned}$$

- 3) *Packet Dropping:* For each link l that has a head-of-line packet that was not successfully transmitted in the scheduling phase (either because its transmission was not attempted, or its transmission failed), drop the packet if $Z_l(t) \leq H_l(t)$. Else, keep it in the head-of-line.

- 4) *Queue Updates:* Update the virtual queues $Z_l(t)$ according to (18), using the values of $\gamma_l(t)$ and $D_l(t)$ as determined from the above auxiliary variable and packet dropping phases. Also update the actual queues and the head-of-line values according to (1) and (20) by simply removing any packet that was either successfully transmitted or dropped.

Theorem 1: Suppose all queues are initially empty, and that Assumption 1 holds. If the above control policy is

implemented with a particular constant $V > 0$, then achieved utility satisfies:

$$\liminf_{t \rightarrow \infty} g(\bar{\mathbf{y}}(t)) \geq g^* - B/V \quad (21)$$

where $\bar{\mathbf{y}}(t)$ is defined in (8) and g^* is the optimal utility. Finally, for all slots t and all links l , we have:

$$Q_l(t) \leq H_l(t) \leq H_{l,max} \triangleq \lceil V\nu_l \rceil + 2, \quad Z_l(t) \leq H_{l,max} \quad (22)$$

The above theorem provides the strong *deterministic guarantee* that head-of-line packets in queue l always have delay less than or equal to $H_{l,max}$, and hence all non-dropped packets in queue l have delay upper bounded by this value. Thus, if we wish to enforce the constraint that worst-case delay in all queues is less than or equal to D_{max} , we can choose V to satisfy $\lceil V \max_l \{\nu_l\} \rceil + 2 = D_{max}$. As the delay constraint D_{max} is relaxed, the value V goes to infinity, and hence by (21) we know that utility converges to the optimal value g^* . The proof of Theorem 1 also provides a more detailed statement regarding the utility achieved over any finite horizon of t slots (see inequality (29) of the next section).

Remark 1: In the case when link reliability is also affected by a set of additional decision parameters $I(t) \in \mathcal{I}$, as discussed in Section II-A, the transmission scheduling decision in phase 2 of the algorithm is modified to maximize:

$$\sum_l x_l(t) \min[H_l(t), Z_l(t)]\Psi_l(\mathbf{x}(t), \mathbf{S}(t), I(t)) \quad (23)$$

subject to $\mathbf{x}(t) \in \mathcal{X}$ and $I(t) \in \mathcal{I}$. Theorem 1 holds exactly as stated under this modification, with the understanding that the optimal utility value g^* may change due to the increased options for scheduling. The machine learning framework of [32] suggests a technique for approximating the reliability function value under the many decision options in \mathcal{I} and \mathcal{X} and under a (possibly approximate) channel state $\mathbf{S}(t)$. Max-weight learning algorithms in [34] also suggest ways of optimizing functionals of the type (23) for unknown environments.

IV. PERFORMANCE ANALYSIS

Here we prove Theorem 1. We first prove the deterministic bound (22), which uses a preliminary lemma.

Lemma 3: If $Z_l(t) > V\nu_l$ for a particular slot t and link l , then the auxiliary variable selection in the first phase of the control algorithm chooses $\gamma_l(t) = -1$ for that slot.

Proof: The value of $\gamma_l(t)$ is determined by maximizing $V\hat{g}(\gamma) - \sum_m Z_m(t)\gamma_m$ over $-1 \leq \gamma \leq 1$. By (12) we know that for any vector γ such that $-1 \leq \gamma \leq 1$:

$$\begin{aligned} &V\hat{g}(\gamma) - \sum_m Z_m(t)\gamma_m \leq \\ &V\hat{g}(\gamma'_l) + V\nu_l(\gamma_l + 1) - \sum_m Z_m(t)\gamma_m \end{aligned}$$

where γ'_l is formed from γ by replacing entry γ_l with -1 . Because $V\nu_l < Z_l(t)$, the right hand side of the above bound is maximized at $\gamma_l = -1$, so that:

$$V\hat{g}(\gamma) - \sum_m Z_m(t)\gamma_m \leq V\hat{g}(\gamma'_l) - \sum_{m \neq l} Z_m(t)\gamma_m + Z_l(t)$$

and equality holds if and only if $\gamma_l = -1$. Hence, the auxiliary variable optimization *must* choose $\gamma_l(t) = -1$. ■

Proof: (Deterministic Bound (22)) Fix a link l . We simultaneously show that $Z_l(t)$ and $H_l(t)$ are both upper bounded by $\lceil V\nu_l \rceil + 2$ for all $t \geq 0$. This clearly holds for $t = 0$ when all queues are empty. Suppose this bound holds for $Z_l(t)$ and $H_l(t)$ at some time t . We prove it also holds for time $t + 1$.

Note from (18) that $Z_l(t)$ can increase by at most 2 every slot (since $D_l(t) \leq 1$ and $\gamma_l(t) \leq 1$ for all t). If $Z_l(t) \leq \lceil V\nu_l \rceil$, then $Z_l(t+1) \leq \lceil V\nu_l \rceil + 2$ and the bound holds. Else, we have $\lceil V\nu_l \rceil < Z_l(t) \leq \lceil V\nu_l \rceil + 2$, and so by Lemma 3 we know that $\gamma_l(t) = -1$. Because $D_l(t) \leq 1$ for all t , we have:

$$D_l(t) + \gamma_l(t) \leq 0 \quad (24)$$

Hence, from the update equation for $Z_l(t+1)$ in (18), we have $Z_l(t+1) \leq Z_l(t) \leq \lceil V\nu_l \rceil + 2$, so the bound again holds.

To show that $H_l(t+1)$ is bounded by the same value, first suppose that $H_l(t) \leq \lceil V\nu_l \rceil + 1$. Because the head-of-line delay can increase by at most 1 every slot, we know that $H_l(t+1) \leq \lceil V\nu_l \rceil + 2$. In the opposite case when $H_l(t) > \lceil V\nu_l \rceil + 1$, we know that $H_l(t) = \lceil V\nu_l \rceil + 2$ (since, by assumption for slot t , $H_l(t)$ must be an integer that is bounded by $\lceil V\nu_l \rceil + 2$). It follows that $H_l(t) \geq Z_l(t)$, and so by the packet dropping procedure in phase 3 of the algorithm, the head-of-line packet will either be successfully transmitted on this slot, or dropped. It follows that in slot $t+1$ there will be either *no* head-of-line packet (so that $H_l(t+1) = 0$), or there will be a *new* head-of-line packet, in which case its delay is no more than the delay of the previous head-of-line packet.

Thus, both $H_l(t+1)$ and $Z_l(t+1)$ are bounded by $\lceil V\nu_l \rceil + 2$. By induction, the result holds for all t . Finally, because there is at most one packet arrival to link l per slot, it is clear that the number of packets in the queue is no more than the current delay of the head-of-line packet, so that $Q_l(t) \leq H_l(t)$. ■

Lemma 4: The control policy chooses decision variables that minimize the right hand side of the drift-minus-utility inequality in Lemma 2.

Proof: See Appendix A. ■

We now prove the utility bound (21). We first use a preliminary lemma, which demonstrates that our structure of making a transmission decision *first*, and then choosing to drop at most one packet per queue, does not limit optimality.

Lemma 5: Let \mathbf{y}^* be the optimal time average throughput vector that solves (9)-(11), so that $g(\mathbf{y}^*) = g^*$, and \mathbf{y}^* satisfies the constraints $\mathbf{y}^* \in \Lambda$ and $\mathbf{0} \leq \mathbf{y}^* \leq \boldsymbol{\lambda}$. Then there is a \mathcal{S} -only algorithm that is independent of the current $\boldsymbol{\Theta}(t)$, and that observes $\mathcal{S}(t)$ and makes a randomized transmission decision $\mathbf{x}^*(t)$ (leading to a vector $\boldsymbol{\mu}^*(t)$ of successful transmissions), and then makes randomized packet drop decisions $\mathbf{D}^*(t)$ in reaction to the ACK/NACK feedback, such that:

$$\mathbb{E}\{\boldsymbol{\mu}^*(t)\} = \mathbf{y}^* \quad , \quad \mathbb{E}\{\mathbf{D}^*(t)\} = \boldsymbol{\lambda} - \mathbf{y}^* \quad (25)$$

Proof: The proof follows easily from the fact (3), and is omitted for brevity. ■

Proof: (Utility Bound (21)) Because our control algorithm satisfies the drift inequality in Lemma 2, and (by Lemma 4) it minimizes the right hand side of this inequality every slot,

we know that:

$$\begin{aligned} \Delta(\boldsymbol{\Theta}(t)) - V\mathbb{E}\{\hat{g}(\boldsymbol{\gamma}(t)) \mid \boldsymbol{\Theta}(t)\} &\leq B - V\mathbb{E}\{\hat{g}(\boldsymbol{\gamma}^*(t)) \mid \boldsymbol{\Theta}(t)\} \\ &\quad - \sum_l Z_l(t)\mathbb{E}\{\lambda_l - D_l^*(t) - \gamma_l^*(t) \mid \boldsymbol{\Theta}(t)\} \\ &\quad - \sum_l H_l(t)\mathbb{E}\{\mu_l^*(t) + D_l^*(t) - \lambda_l \mid \boldsymbol{\Theta}(t)\} \end{aligned}$$

where $\boldsymbol{\gamma}^*(t)$, $\mathbf{D}^*(t)$, and $\boldsymbol{\mu}^*(t)$ are from any \mathcal{S} -only policy (independent of $\boldsymbol{\Theta}(t)$). Taking expectations of both sides of the above and using the law of iterated expectations yields:

$$\begin{aligned} \mathbb{E}\{L(\boldsymbol{\Theta}(t+1))\} - \mathbb{E}\{L(\boldsymbol{\Theta}(t))\} - V\mathbb{E}\{\hat{g}(\boldsymbol{\gamma}(t))\} &\leq \\ &\quad B - V\mathbb{E}\{\hat{g}(\boldsymbol{\gamma}^*(t))\} \\ &\quad - \sum_l \mathbb{E}\{Z_l(t)\} \mathbb{E}\{\lambda_l - D_l^*(t) - \gamma_l^*(t)\} \\ &\quad - \sum_l \mathbb{E}\{H_l(t)\} \mathbb{E}\{\mu_l^*(t) + D_l^*(t) - \lambda_l\} \quad (26) \end{aligned}$$

Now choose the alternative auxiliary variable decision:

$$\boldsymbol{\gamma}^*(t) = \mathbf{y}^* \quad (27)$$

This is a feasible decision because the optimal vector \mathbf{y}^* satisfies $0 \leq y_l^* \leq 1$ for all l , and so clearly $-1 \leq \gamma_l^* \leq 1$ for all l . Further, choose the alternative decisions for $\mathbf{D}^*(t)$ and $\boldsymbol{\mu}^*(t)$ that are the \mathcal{S} -only decisions that yield (25) from Lemma 5. Plugging (27) and (25) into the right hand side of (26) and using the fact that $g(\mathbf{y}^*) = g^*$ yields:

$$\mathbb{E}\{L(\boldsymbol{\Theta}(t+1))\} - \mathbb{E}\{L(\boldsymbol{\Theta}(t))\} - V\mathbb{E}\{\hat{g}(\boldsymbol{\gamma}(t))\} \leq B - Vg^*$$

The above holds for all $t \geq 0$. Summing over $\tau \in \{0, \dots, t-1\}$ and dividing by t yields:

$$\frac{\mathbb{E}\{L(\boldsymbol{\Theta}(t))\} - \mathbb{E}\{L(\boldsymbol{\Theta}(0))\}}{t} - \frac{V}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{\hat{g}(\boldsymbol{\gamma}(\tau))\} \leq B - Vg^*$$

Using the fact that $L(\cdot) \geq 0$ and rearranging terms yields:

$$\frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{\hat{g}(\boldsymbol{\gamma}(\tau))\} \geq g^* - B/V - \frac{\mathbb{E}\{L(\boldsymbol{\Theta}(0))\}}{Vt}$$

Using Jensen's inequality in the concave function $\hat{g}(\cdot)$ yields:

$$\hat{g}(\bar{\boldsymbol{\gamma}}(t)) \geq g^* - B/V - \frac{\mathbb{E}\{L(\boldsymbol{\Theta}(0))\}}{Vt} \quad (28)$$

where $\bar{\boldsymbol{\gamma}}(t)$ is defined as:

$$\bar{\boldsymbol{\gamma}}(t) \triangleq \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{\boldsymbol{\gamma}(\tau)\}$$

However, because each $Z_l(t)$ queue is deterministically bounded by the finite constant $H_{l,max}$, we have from (19):

$$\bar{\mathbf{y}}(t) + \mathbf{H}_{max}/t \geq \bar{\boldsymbol{\gamma}}(t)$$

where $\mathbf{H}_{max} = (H_{l,max})_{l=1}^L$. Plugging this into (28) and using the fact that $\hat{g}(\cdot)$ is non-decreasing in each entry yields:

$$\hat{g}(\bar{\mathbf{y}}(t) + \mathbf{H}_{max}/t) \geq g^* - B/V - \frac{\mathbb{E}\{L(\boldsymbol{\Theta}(0))\}}{Vt} \quad (29)$$

The above holds for all t . By continuity of $\hat{g}(\cdot)$ and the fact that $H_{l,max}/t \rightarrow 0$, we have:

$$\liminf_{t \rightarrow \infty} \hat{g}(\bar{\mathbf{y}}(t)) \geq g^* - B/V \quad (30)$$

Using the definition of $\bar{\mathbf{y}}(t)$ in (8) it can be shown that for all l and all t we have $0 \leq \bar{y}_l(t) \leq 1$. Because $g(\mathbf{y}) = \hat{g}(\mathbf{y})$ whenever $\mathbf{0} \leq \mathbf{y} \leq \mathbf{1}$, we have:

$$\liminf_{t \rightarrow \infty} \hat{g}(\bar{\mathbf{y}}(t)) = \liminf_{t \rightarrow \infty} g(\bar{\mathbf{y}}(t))$$

Using this in (30) proves (21). \blacksquare

V. CONCLUSION

We have established a delay-based policy for joint stability and utility optimization. The policy provides deterministic worst-case delay bounds, with total throughput-utility that is inversely proportional to the delay guarantee. The Lyapunov Optimization approach for this delay-based problem is significantly different from that of backlog-based policies. Further, delay-based scheduling must overcome difficult issues involving the correlation between inter-arrival times and virtual queue states. Several new techniques were introduced to solve the problem, including the structure of dropping packets at the head-of-line (rather than immediately upon arrival), introducing the concept of concavely extending a utility function, and using a delayed arrival process $A_l(t - W)$ in the virtual queues to maintain required independence. We believe these results add significantly to our understanding of network delay and delay-efficient control laws.

APPENDIX A — PROOF OF LEMMA 4

Here we show that the given delay-based control algorithm maximizes the following expression, which is an expression that considers only the terms in the right hand side of the drift bound in Lemma 2 that involve control parameters:

$$V \mathbb{E} \{ \hat{g}(\gamma(t)) \mid \Theta(t) \} - \sum_l Z_l(t) \mathbb{E} \{ D_l(t) + \gamma_l(t) \mid \Theta(t) \} \\ + \sum_l H_l(t) \mathbb{E} \{ \mu_l(t) + D_l(t) \mid \Theta(t) \}$$

Note that the $\gamma_l(t)$ terms appear separably in this drift expression, and hence they can be optimally chosen by observing $\Theta(t)$ and maximizing:

$$V \hat{g}(\gamma) - \sum_l Z_l(t) \gamma_l(t)$$

subject to $-1 \leq \gamma_l(t) \leq 1$ for all links l . This is precisely the first phase of the control algorithm.

Note that the remaining terms can be rearranged as:

$$\sum_l H_l(t) \mathbb{E} \{ \mu_l(t) \mid \Theta(t) \} + \sum_l (H_l(t) - Z_l(t)) \mathbb{E} \{ D_l(t) \mid \Theta(t) \}$$

Define $m_l(t)$ as an indicator function that is 1 if $H_l(t) \geq Z_l(t)$, and 0 else. It is clear that if $m_l(t) = 1$ and if there is a head-of-line packet in link l , then this packet should be dropped (if it is not already transmitted successfully), and these should *not* be dropped if $m_l(t) = 0$. This is exactly the packet-dropping phase (phase 3) in the control algorithm.

It now suffices to choose an optimal transmission vector $\mathbf{x}(t)$, where $x_l(t) \in \{0, 1\}$. Recall that $\mu_l(t) = x_l(t)1_l(t)$, where $1_l(t)$ is an indicator function that is 1 if and only if the packet in link l was transmitted successfully. Noting that

$D_l(t) = 1$ only for $m_l(t) = 1$, the above expression is thus (written without the conditional expectation for convenience):

$$\sum_l H_l(t) x_l(t) 1_l(t) + \sum_l (H_l(t) - Z_l(t)) D_l(t) x_l(t) m_l(t) \\ + \sum_l D_l(t) (H_l(t) - Z_l(t)) (1 - x_l(t)) m_l(t) \\ = \sum_l H_l(t) x_l(t) 1_l(t) \\ + \sum_l (H_l(t) - Z_l(t)) (1 - 1_l(t)) x_l(t) m_l(t) \\ + \sum_l (H_l(t) - Z_l(t)) (1 - x_l(t)) m_l(t)$$

where we have used the fact that if $x_l(t) = 1$ and $m_l(t) = 1$, then $D_l(t) = 1$ if and only if $1_l(t) = 0$ (so that if a packet for which $m_l(t) = 1$ is transmitted unsuccessfully, it is necessarily dropped). Likewise, we have used the fact that if $x_l(t) = 0$ and $m_l(t) = 1$, then $D_l(t) = 1$.

Rearranging terms of the above that involve control decisions yields:

$$\sum_l x_l(t) [H_l(t) 1_l(t) - m_l(t) 1_l(t) (H_l(t) - Z_l(t))]$$

Further rearrangements yield:

$$\sum_l x_l(t) 1_l(t) [H_l(t) - m_l(t) (H_l(t) - Z_l(t))] \quad (31)$$

However, we have:

$$H_l(t) - m_l(t) (H_l(t) - Z_l(t)) = \min[H_l(t), Z_l(t)]$$

This is because $m_l(t) = 1$ if and only if $H_l(t) \geq Z_l(t)$, so that $m_l(t) = 1$ implies the expression is $H_l(t) - (H_l(t) - Z_l(t)) = Z_l(t) = \min[H_l(t), Z_l(t)]$, and likewise $m_l(t) = 0$ implies the expression is $H_l(t) = \min[H_l(t), Z_l(t)]$. Plugging this identity into the expression (31) yields:

$$\sum_l x_l(t) 1_l(t) \min[H_l(t), Z_l(t)]$$

Taking conditional expectations of the above with respect to $\Theta(t)$ yields:

$$\mathbb{E} \left\{ \sum_l x_l(t) 1_l(t) \min[H_l(t), Z_l(t)] \mid \Theta(t) \right\}$$

We seek a control rule that observes $\mathcal{S}(t)$ and $\Theta(t)$ and chooses $\mathbf{x}(t) \in \mathcal{X}$, so that the above expression is maximized. Define $\chi(t) = [\mathcal{S}(t), \Theta(t), \mathbf{x}(t)]$. By iterated expectations the above expression is:

$$\mathbb{E} \left\{ \mathbb{E} \left\{ \sum_l x_l(t) 1_l(t) \min[H_l(t), Z_l(t)] \mid \chi(t) \right\} \mid \Theta(t) \right\} \\ = \mathbb{E} \left\{ \sum_l x_l(t) \Psi_l(\mathbf{x}(t), \mathcal{S}(t)) \min[H_l(t), Z_l(t)] \mid \Theta(t) \right\}$$

where we have used the fact that $\mathbb{E} \{ 1_l(t) \mid \chi(t) \} = \Psi(\mathbf{x}(t), \mathcal{S}(t))$. The above expectation is thus minimized by observing the current $\mathcal{S}(t)$ and allocating $\mathbf{x}(t) \in \mathcal{X}$ according to phase 2 of the control algorithm.

REFERENCES

- [1] L. Georgiadis, M. J. Neely, and L. Tassiulas. Resource allocation and cross-layer control in wireless networks. *Foundations and Trends in Networking*, vol. 1, no. 1, pp. 1-149, 2006.
- [2] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936-1949, Dec. 1992.
- [3] L. Tassiulas and A. Ephremides. Dynamic server allocation to parallel queues with randomly varying connectivity. *IEEE Transactions on Information Theory*, vol. 39, pp. 466-478, March 1993.
- [4] N. McKeown, V. Anantharam, and J. Walrand. Achieving 100% throughput in an input-queued switch. *Proc. IEEE INFOCOM*, 1996.
- [5] P. R. Kumar and S. P. Meyn. Stability of queueing networks and scheduling policies. *IEEE Trans. on Automatic Control*, vol.40.,n.2, pp.251-260, Feb. 1995.
- [6] E. Leonardi, M. Mellia, F. Neri, and M. Ajmone Marsan. Bounds on average delays and queue size averages and variances in input-queued cell-based switches. *Proc. IEEE INFOCOM*, 2001.
- [7] N. Kahale and P. E. Wright. Dynamic global packet routing in wireless networks. *Proc. IEEE INFOCOM*, 1997.
- [8] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, and P. Whiting. Providing quality of service over a shared wireless link. *IEEE Communications Magazine*, vol. 39, no.2, pp.150-154, 2001.
- [9] M. J. Neely, E. Modiano, and C. E Rohrs. Dynamic power allocation and routing for time varying wireless networks. *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 89-103, January 2005.
- [10] M. Kobayashi, G. Caire, and D. Gesbert. Impact of multiple transmit antennas in a queued SDMA/TDMA downlink. In *Proc. of 6th IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, June 2005.
- [11] M. J. Neely and R. Uргаonkar. Optimal backpressure routing in wireless networks with multi-receiver diversity. *Ad Hoc Networks (Elsevier)*, vol. 7, no. 5, pp. 862-881, July 2009.
- [12] M. J. Neely, E. Modiano, and C. Li. Fairness and optimal stochastic control for heterogeneous networks. *Proc. IEEE INFOCOM*, March 2005.
- [13] M. J. Neely. *Dynamic Power Allocation and Routing for Satellite and Wireless Networks with Time Varying Channels*. PhD thesis, Massachusetts Institute of Technology, LIDS, 2003.
- [14] A. Stolyar. Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm. *Queueing Systems*, vol. 50, pp. 401-457, 2005.
- [15] X. Lin and N. B. Shroff. Joint rate control and scheduling in multihop wireless networks. *Proc. of 43rd IEEE Conf. on Decision and Control, Paradise Island, Bahamas*, Dec. 2004.
- [16] A. Eryilmaz and R. Srikant. Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control. *Proc. IEEE INFOCOM*, March 2005.
- [17] J. W. Lee, R. R. Mazumdar, and N. B. Shroff. Opportunistic power scheduling for dynamic multiserver wireless systems. *IEEE Transactions on Wireless Communications*, vol. 5, no.6, pp. 1506-1515, June 2006.
- [18] R. Agrawal and V. Subramanian. Optimality of certain channel aware scheduling policies. *Proc. 40th Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL*, Oct. 2002.
- [19] H. Kushner and P. Whiting. Asymptotic properties of proportional-fair sharing algorithms. *Proc. of 40th Annual Allerton Conf. on Communication, Control, and Computing*, 2002.
- [20] A. Mekikittikul and N. McKeown. A starvation free algorithm for achieving 100% throughput in an input-queued switch. *Proc. ICCN*, pp. 226-231, 1996.
- [21] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijaykumar, and P. Whiting. Scheduling in a queueing system with asynchronously varying service rates. *Probability in the Engineering and Informational Sciences*, vol. 18, no. 2, pp. 191-217, April 2004.
- [22] S. Shakkottai and A. Stolyar. Scheduling multiple flows sharing a time-varying channel: The exponential rule. *American Mathematical Society Translations, series 2*, vol. 207, 2002.
- [23] M. J. Neely and R. Uргаonkar. Cross layer adaptive control for wireless mesh networks. *Ad Hoc Networks (Elsevier)*, vol. 5, no. 6, pp. 719-743, August 2007.
- [24] M. J. Neely. Energy optimal control for time varying wireless networks. *IEEE Transactions on Information Theory*, vol. 52, no. 7, pp. 2915-2934, July 2006.
- [25] R. Berry and R. Gallager. Communication over fading channels with delay constraints. *IEEE Transactions on Information Theory*, vol. 48, no. 5, pp. 1135-1149, May 2002.
- [26] M. J. Neely. Super-fast delay tradeoffs for utility optimal fair scheduling in wireless networks. *IEEE Journal on Selected Areas in Communications, Special Issue on Nonlinear Optimization of Communication Systems*, vol. 24, no. 8, pp. 1489-1501, Aug. 2006.
- [27] D. P. Bertsekas and R. Gallager. *Data Networks*. New Jersey: Prentice-Hall, Inc., 1992.
- [28] B. Sadiq, S. Baek, and Gustavo de Veciana. Delay-optimal opportunistic scheduling and approximations: the log rule. *Proc. IEEE INFOCOM*, April 2009.
- [29] M. J. Neely. Stochastic optimization for markov modulated networks with application to delay constrained wireless scheduling. *Proc. IEEE Conf. on Decision and Control (CDC)*, Shanghai, China, Dec. 2009.
- [30] J. K. MacKie-Mason and H. R. Varian. Pricing congestible network resources. *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 7, September 1995.
- [31] M. J. Neely and E. Modiano. Convexity in queues with general inputs. *IEEE Transactions on Information Theory*, vol. 51, no. 2, pp. 706-714, Feb. 2005.
- [32] R. Daniels and R. W. Heath Jr. An online learning framework for link adaptation in wireless networks. *Proc. Information Theory and Applications Workshop, San Deigo, CA*, 2009.
- [33] F. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, vol. 8, pp. 33-37, 1997.
- [34] M. J. Neely. Max weight learning algorithms with application to scheduling in unknown environments. *arXiv:0902.0630v1*, Feb. 2009.