Super-Fast Delay Tradeoffs for Utility Optimal Fair Scheduling in Wireless Networks

Michael J. Neely
University of Southern California
http://www-rcf.usc.edu/∼mjneely

***Abstract*— We consider the fundamental delay tradeoffs for utility optimal scheduling in a general multi-hop network with time varying channels. A network controller acts on randomly arriving data and makes flow control, routing, and resource allocation decisions to maximize a fairness metric based on a concave utility function of network throughput. A simple set of algorithms are constructed that yield total utility within $O(1/V)$ of the utility-optimal operating point, for any control parameter $V > 0$, with a corresponding end-to-end network delay that grows only logarithmically in V. This is the first algorithm to achieve such "super-fast" performance. Furthermore, we show that this is the best utility-delay tradeoff possible. This work demonstrates that the problem of maximizing throughput utility in a data network is fundamentally different than related problems of minimizing average power expenditure, as these latter problems cannot achieve such performance tradeoffs.**

***Index Terms*— Fairness, flow control, wireless networks, queueing analysis, optimization, delay, network capacity**

I. Introduction

We consider the fundamental tradeoff between network utility and network delay for a general multi-hop network with both wireless and wireline components and time varying channels. Traffic arrives to the network randomly, and we assume input rates exceed network capacity. Such a situation is typical for modern networks where growing user demands can quickly overload physical system resources. It is essential to establish simple solutions that maintain low network congestion and delay while providing fair access to all users. In this paper, we evaluate fairness according to a general concave utility function of the long term throughput of each user. The goal is to design a controller that drives total utility towards its maximum value, considering all possible methods of flow control, routing, and resource allocation, while ensuring an optimal tradeoff in end-to-end network delay.

In our previous work on the network fairness problem, we constructed a set of algorithms indexed by a control parameter $V > 0$ that yield total network utility within $O(1/V)$ of the utility-optimal operating point, with a corresponding delay tradeoff that is linear in V [1] [2]. This result suggests that delay necessarily increases when utility is pushed toward optimality, although the existence of such a tradeoff and the form of the optimal utility-delay curve were left as open questions. In this paper we explore these questions and characterize the fundamental tradeoff curve. Specifically, we consider an overloaded system where user data rates strictly dominate the optimally fair operating point. We then develop a novel algorithm that deviates from the optimal utility by no more than $O(1/V)$ while ensuring that average end-to-end network delay is less than or equal to $O(\log(V))$. Further, we prove that no algorithm can achieve a better asymptotic tradeoff. This establishes a fundamental relationship between utility and delay, and demonstrates the unexpected result that logarithmic delay is possible for systems with any concave utility metric.

This material is based upon work supported in part by the National Science Foundation under grant OCE 0520324.

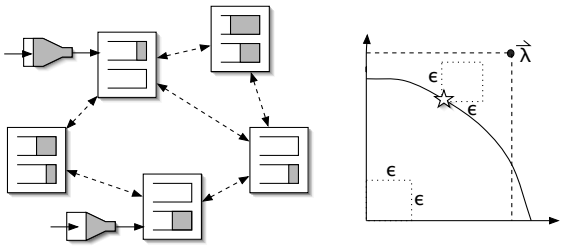

Fig. 1. An example network of 5 nodes, and an illustration of a capacity region (shown in two dimensions) with an input rate vector that strictly dominates the optimal operating point.

Related work in [4] considers the tradeoff between energy and delay for a single queue that stores data for transmission over a single fading channel. There, it is shown that any scheduling policy yielding average energy expenditure within $O(1/V)$ of the minimum energy required for stability must also have average queueing delay greater than or equal to $\Omega(\sqrt{V})$. Strategies for achieving this tradeoff are proposed in [4] using the concept of *buffer partitioning*, and a recent result in [32] shows that the same square-root tradeoff applies to the minimum energy problem for multi-user networks.

In this paper, we combine the technique of buffer partitioning with the recently developed technique of *performance optimal Lyapunov scheduling* [1] [2] [3]. Specifically, in [1] [2] [3] a Lyapunov drift theorem is developed that treats stability and performance optimization simultaneously, leading to simple and robust control strategies. Here, we extend the theory to treat optimal utility-delay tradeoffs. The result is a novel set of Lyapunov scheduling algorithms that can be used for any network, without requiring a-priori knowledge of traffic rates or channel statistics. The algorithms use weights that aggressively switch ON and OFF in order to achieve optimal delay tradeoffs. We find that the special structure of the long-term fairness objective allows for a "super-fast" logarithmic delay tradeoff that cannot be achieved in related problems of minimizing energy expenditure.

It is important to distinguish the tradeoffs we explore here to the capacity-delay tradeoffs recently explored for ad-hoc mobile networks in [1], [5]-[9]. These tradeoffs are quite different,

in that they consider the "opposite end" of the performance curve. Indeed, in this paper we consider the impact of pushing network utility arbitrarily close to optimal, whereas the work in [1], [5]-[9] considers the opposite scenario where network utility (which is measured by throughput) is dramatically reduced with the goal of also reducing network delay. These antipodal ends of the tradeoff curve are conceptually different and involve completely different analytical methods.

Previous work in the area of utility-optimal scheduling is closely tied to the theory of convex optimization and Lagrangian duality. In [10], a network flow problem is formulated in terms of a network utility function, and a technique of introducing Lagrange multipliers as "shadow prices" was shown to offer distributed solution strategies. The relationship between convex duality theory and TCP-like congestion control algorithms is explored in [11]. Convex optimization approaches to static wireless network problems are considered in [12]-[20]. Stability problems for stochastic networks are treated in [21]-[27] using Lyapunov stability theory, and recent approaches to stability and performance optimization are considered in [28]-[31] using fluid models and/or stochastic gradient algorithms to transform a stochastic problem into one that is similar to a static problem. A detailed comparison between static gradient algorithms and stable Lyapunov scheduling is presented in [26] [1], and a Lyapunov method for performance optimization is developed in [1] [2] [3] that yields results similar to those of stochastic gradient algorithms and also provides explicit performance and delay bounds.

The stochastic scheduling techniques we use in this paper are quite new, and go beyond the gradient algorithms suggested by classical optimization theory. Indeed, the techniques even suggest improved approaches to these classical problems. The resulting utility-delay tradeoff that we achieve extends the field of stochastic optimal networking and demonstrates that significant performance gains are possible through simple scheduling policies.

An outline of this paper is as follows: In the next section we introduce the system model. In Sections III and IV we specify the control objective and design the network control algorithm. Optimality of our $O(\log(V))$ delay result is proven in Section V for the special case of one-hop networks. Extensions to cost-constrained networking and open questions concerning optimal decay exponents are considered in Section VI.

II. Problem Formulation

Consider a data network with N nodes, as in Fig. 1. The network operates in slotted time with timeslots $t \in \{0, 1, 2, \ldots\}$, and every timeslot data randomly enters the network at each node. We define the arrival process $A_{nc}(t)$ as the amount of new bits that exogenously enter node n during timeslot t that must be delivered to node c. For simplicity of exposition, we assume arrivals are i.i.d. over timeslots, with arrival rates $\lambda_{nc} = \mathbb{E}\{A_{nc}(t)\}$. These arrival rates are not necessarily known to the network controller. Let $\underline{\lambda} = (\lambda_{nc})$ represent the $N \times N$ matrix of arrival rates. We say that an arrival stream $A_{nc}(t)$ is *active* if $\lambda_{nc} > 0$. Let $\mathbb{M}$ represent the set of all (n, c) pairs associated with active arrival streams. We assume that $A_{nc}(t) = 0$ for all t whenever $(n, c) \notin \mathbb{M}$. Note in particular that $A_{nn}(t) = 0$ for all nodes n and for all time, as no node receives exogenous data destined for itself.

A. Flow Control

Data is not immediately admitted into the network. Rather, we assume that data from stream $A_{nc}(t)$ is first placed into a *storage reservoir* (n, c) at node n. Define $L_{nc}(t)$ as the amount of data currently stored in reservoir (n, c). Every timeslot, a network controller at node n makes *flow control decisions* by choosing $R_{nc}(t)$, the amount of type (n, c) data allowed to enter the network on slot t. Note that $R_{nc}(t) \leq A_{nc}(t) + L_{nc}(t)$. Any newly arriving data that is not admitted to the network is placed into the storage reservoir, or dropped if there is no room for storage. The storage buffer can either be infinite (so that no data is ever dropped), or finite (possibly zero). In the special case of a "size zero" reservoir, any data that is not immediately admitted to the network is necessarily dropped.

B. Resource and Rate Allocation

Nodes transfer data by transmitting over communication links, and we let $\mu_{ab}(t)$ represent the rate offered for transmission from node a to node b during slot t. If there is a wireline data link connecting nodes a and b that can support a rate of γ bits/slot, then $\mu_{ab}(t)$ is equal to the constant γ for all slots t (so that γ bits can be transfered in a single timeslot). However, in wireless systems the transmission rate might depend on resource allocation decisions (such as bandwidth or power allocation), and on time varying and uncontrollable channel conditions (due to environmental effects, fading, or user mobility). Hence, for a general system we define $\underline{S}(t) = (S_{ab}(t))$ as the *channel state matrix* at time t, representing the uncontrollable properties of the channel that effect transmission. We assume that the number of channel states is finite, and that channel state matrices are i.i.d. over timeslots.[1] For each state matrix $\underline{S}$, define $\Omega_{\underline{S}}$ as the compact set of all feasible transmission rate matrices (μ_{ab}) available for resource allocation decisions when $\underline{S}(t) = \underline{S}$. Every timeslot, the network controller observes the current channel state matrix $\underline{S}(t)$ and chooses a transmission rate matrix $(\mu_{ab}(t)) \in \Omega_{\underline{S}(t)}$.

In networks with general inter-channel interference properties, this control decision may require full coordination of all nodes. However, in cases where channels can be decoupled into a collection K of independent sets, we have:

$$\Omega_{\underline{S}} = \Omega^1_{\underline{S}^1} \times \ldots \times \Omega^K_{\underline{S}^K} \tag{1}$$

where $\underline{S}^k$ represents the channel states of data links within the kth independent set. In such a network, resource allocation decisions can be made separately within each set. Alternatively, if a distributed multiple access structure is specified in advance, then subsets $\tilde{\Omega}_{\underline{S}}$ can be defined to represent the effective rate options available under distributed scheduling, which are generally smaller than the sets of rates $\Omega_{\underline{S}}$ available through centralized coordination.

[1] Extensions to non-i.i.d. systems can be treated via the methods in [1] [26].

C. Routing and Network Queueing

We define all data destined for a particular node $c \in \{1, \ldots, N\}$ to be *commodity c data*, regardless of its source. Data can be transfered over the network over multi-hop paths, and hence each node maintains a set of internal queues for storing data according to its destination. Let the "unfinished work" process $U_n^{(c)}(t)$ represent the amount of commodity c data currently stored in node n. Every timeslot, the network controllers at every node must decide which commodities should be transmitted, and which nodes they should be transmitted to. It is often useful to restrict routing options so that data follows a particular path or set of paths to the destination. To enforce this constraint, for each commodity $c \in \{1, \ldots, N\}$ we define $\mathbb{L}_c$ as the set of all data links (a, b) that are acceptable for commodity c data to traverse. In general, multiple commodities could be transmitted over the same data link during a single timeslot.[2] We thus define $\mu_{ab}^{(c)}(t)$ as the transmission rate offered to commodity c data over link (a, b) during timeslot t. Network controllers make *routing decisions* by choosing the $\mu_{ab}^{(c)}(t)$ variables according to the following constraints:

$$\sum_c \mu_{ab}^{(c)}(t) \leq \mu_{ab}(t) \tag{2}$$

$$\mu_{ab}^{(c)}(t) = 0 \quad \text{if } (a, b) \notin \mathbb{L}_c \tag{3}$$

Note that the routing decisions are coupled to the resource allocation decisions, in that the rate offered for routing data over a particular link (a, b) must be less than or equal to the chosen transmission rate $\mu_{ab}(t)$. However, in order to preserve the "layering" concept of data networks, it is useful to think of the routing decisions separately.

Considering the combined effect of flow control, routing, and resource allocation decisions, the queue backlogs change every timeslot according to the following dynamic inequality:

$$U_n^{(c)}(t+1) \leq \max[U_n^{(c)}(t) - \sum_b \mu_{nb}^{(c)}(t), 0] + R_{nc}(t) + \sum_a \mu_{an}^{(c)}(t) \tag{4}$$

This dynamic expression is due to the fact that the backlog of commodity c in a particular node at time $t+1$ is equal to the backlog at time t minus the amount transmitted to other nodes during this timeslot, plus the total arrivals to this node from both the exogenous flow control decisions and the endogenous transmissions from other nodes in the network. This is expressed as an inequality in (4) due to the fact that the total endogenous arrivals may be less than $\sum_a \mu_{an}^{(c)}(t)$ if the other nodes do not have enough commodity c data in their queues to send at the full transmission rate.

III. Network Capacity and the Control Objective

The *network capacity region* Λ is defined as the closure of all arrival rate matrices (λ_{nc}) that the network can stably support. Specifically, if we assume that the flow controllers allow all arriving data directly into the network, then $(\lambda_{nc}) \notin \Lambda$ implies that no routing and resource allocation algorithm that meets the system constraints specified in the previous section can stabilize all queues of the network. However, if (λ_{nc}) is strictly interior to Λ, then there must exist an algorithm that stabilizes the network and thereby achieves a throughput matrix equal to the input rate matrix (λ_{nc}).

In [1] [26], the capacity region Λ is considered for the special case when the routing constraints (3) are removed. This region can be understood by considering a set of *multi-commodity flow variables* that specify routes to destinations, together with a stationary randomized resource allocation policy that chooses transmission rates in reaction to current channel states but independently of current queue backlog. It was shown that the capacity region Λ is compact and convex, and consists of all rate matrices (λ_{nc}) for which there exists a multi-commodity flow together with a stationary randomized resource allocation policy such that the flow rate over every link is less than or equal to the time average transmission rate over that link. This result can be easily extended to include the routing constraints (3). We thus state the following modified version of the network capacity theorem from [1] [26], omitting the proof for brevity.

Theorem 1: (Network Capacity with Routing Constraints) An input rate matrix (λ_{nc}) is in the capacity region Λ if and only if there exists a stationary randomized resource allocation algorithm that chooses transmission rates $\mu_{ab}^{(c)}(t)$ based only on the current channel state, and satisfies for all slots t:

$$\mathbb{E}\left\{\sum_b \mu_{nb}^{(c)}(t) - \sum_a \mu_{an}^{(c)}(t)\right\} = \lambda_{nc} \quad \text{for all } n \neq c \tag{5}$$

$$\mu_{ab}^{(a)}(t) = 0 \quad \text{for all } (a, b)$$

$$\mu_{ab}^{(c)}(t) = 0 \quad \text{for all } (a, b, c) \text{ such that } (a, b) \notin \mathbb{L}_c$$

The expectation in (5) is taken over the probability distribution of the current channel state $\underline{S}(t)$ and the distribution of the randomized resource allocation decisions that depend on this channel state. Note that because channel states are i.i.d. from slot to slot, the above expectation is the same every timeslot, and does not depend on the channel state history from previous slots. The above theorem can be intuitively understood by associating multi-commodity flow variables with the expected link transmission rates as follows:

$$f_{ab}^{(c)} \triangleq \mathbb{E}\left\{\mu_{ab}^{(c)}(t)\right\}$$

The constraint (5) can thus be viewed as a *flow conservation constraint*, specifying that the total in-flow of commodity c data into a node must be equal to the total out-flow, provided that the node is not the destination of commodity c.

A. The Control Objective

Let r_{nc} represent the long term rate that data from stream $A_{nc}(t)$ is delivered to the network by the flow controller at node n. Clearly $r_{nc} \leq \lambda_{nc}$ for all (n, c). Further, to ensure network stability we must have $(r_{nc}) \in \Lambda$. As a conventional measure of the total utility associated with supporting a traffic rate r_{nc}, for each node pair (n, c) we define a *utility function*

[2] We shall find that, without loss of optimality, we can restrict data links to transmitting only one commodity per slot.

$g_{nc}(r)$. Utility functions are assumed to be non-negative, non-decreasing, concave, and to have the property that $g_{nc}(0) = 0$ for all (n, c) (so that zero throughput for a given stream implies zero utility for that stream). Such utility functions are often used as a quantitative measure of *network fairness* [33] [10] [18] [16] [13] [29] [2] [1], and different choices of $g_{nc}(r)$ lead to different fairness properties [34]. We further assume that all utility functions have finite right derivatives. The *optimal network utility* g^* is defined as the solution of the following problem:

$$\begin{aligned} \text{Maximize:} \quad & \textstyle\sum_{nc} g_{nc}(r_{nc}) \\ \text{Subject to:} \quad & (r_{nc}) \in \Lambda \\ & r_{nc} \leq \lambda_{nc} \quad \text{for all } (n,c) \end{aligned} \tag{6}$$

Note that no control algorithm that stabilizes the network can achieve an overall utility larger than g^*. The goal is to develop a joint algorithm for flow control, routing, and resource allocation that stabilizes the network and yields a total utility that can be pushed arbitrarily close to the optimal utility g^*, with a corresponding optimal tradeoff in end-to-end network delay.

Because the capacity region Λ is compact, there exists a (potentially non-unique) optimal rate matrix (r^*_{nc}) such that $\sum_{nc} g_{nc}(r^*_{nc}) = g^*$. We refer to such a rate matrix as an *optimally fair operating point*. Note that if (λ_{nc}) is strictly interior to the capacity region, then $(r^*_{nc}) = (\lambda_{nc})$, and hence the exact optimal utility can be achieved simply by stabilizing the network. This can be accomplished with finite average delay using the DRPC algorithm of [1] [26]. Hence, the notion of a fundamental utility-delay tradeoff only makes sense in the case when the input rate matrix is either on the boundary or strictly outside of the capacity region Λ.

In this paper, we focus on the case when the rate matrix is strictly outside of Λ. More precisely, we shall assume there exists a positive value ϵ_{max} and an optimally fair operating point (r^*_{nc}) such that $\lambda_{nc} \geq r^*_{nc} + \epsilon_{max}$ for all active sessions $(n, c) \in \mathbb{M}$. Such a scenario occurs, for example, when all active sessions have infinite storage reservoirs that are infinitely backlogged so that there is always data to send (as in [29] [28] [18] [10] [20]), or when input rates are sufficiently large so that the matrix (λ_{nc}) dominates the optimally fair operating point (r^*_{nc}) in each entry $(n, c) \in \mathbb{M}$ (see Fig. 1). Under this assumption, in the next section we show that a "super-fast" logarithmic delay tradeoff is possible, improving upon the linear tradeoff shown to be achievable for all rate matrices in [2]. If the input rates are outside of capacity but do not strictly dominate the optimally fair operating point, then it must be that $r^*_{nc} = \lambda_{nc}$ for at least one node pair $(n, c) \in \mathbb{M}$. We conjecture that such singularities preclude "super-fast" logarithmic tradeoffs.

IV. The Dynamic Control Algorithm

To motivate the control algorithm, first note that the optimization problem (6) is equivalent to the following:

$$\text{Maximize:} \quad \textstyle\sum_{nc} g_{nc}(\gamma_{nc}) \tag{7}$$

$$\text{Subject to:} \quad r_{nc} \geq \gamma_{nc} \quad \text{for all } (n,c) \tag{8}$$

$$(r_{nc}) \in \Lambda \tag{9}$$

$$r_{nc} \leq \lambda_{nc} \quad \text{for all } (n,c) \tag{10}$$

The additional linear constraint (8) acts to decouple the throughput values (r_{nc}) from the new optimization variables (γ_{nc}). To build intuition, suppose we have a network algorithm controlling the system that stabilizes all queues and yields well defined time averages for throughput and transmission rates. Note that any such algorithm will have a throughput matrix (r_{nc}) that automatically satisfies (9) and (10). Now define $\overline{\mu}^{(c)}_{ab}$ as the time average transmission rate offered to commodity c data over link (a, b), and let $\hat{\mu}^{(c)}_{ab}$ represent the time average rate that actual commodity c data is delivered over link (a, b) (assuming for now that such time averages exist). Note that $\hat{\mu}^{(c)}_{ab} \leq \overline{\mu}^{(c)}_{ab}$, where strict inequality is possible due to "edge effects" that arise when the queue backlog $U^{(c)}_a(t)$ is frequently zero or near-zero so that the offered transmission rate $\mu^{(c)}_{ab}(t)$ is under-utilized. The time average departure rate of commodity c bits from node n must be less than or equal to the maximum possible sum rate of endogenous and exogenous bits into this node, and hence:

$$r_{nc} + \sum_a \overline{\mu}^{(c)}_{an} \geq \sum_b \hat{\mu}^{(c)}_{nb}$$

However, if we can by some means ensure that edge effects arise very infrequently, then $\hat{\mu}^{(c)}_{nb} \approx \overline{\mu}^{(c)}_{nb}$, and hence for some small value $\delta > 0$ we have:

$$r_{nc} + \delta \geq \sum_b \overline{\mu}^{(c)}_{nb} - \sum_a \overline{\mu}^{(c)}_{an} \tag{11}$$

Hence, if we can design a stabilizing control algorithm that maximizes (7), keeps queues far away from the edge region, and that satisfies the following constraints for all (n, c):

$$\sum_b \overline{\mu}^{(c)}_{nb} - \sum_a \overline{\mu}^{(c)}_{an} \quad \geq \quad \gamma_{nc} \tag{12}$$

then from (11) we have that the constraint (8) will be within δ of being satisfied, and so the resulting network utility will be close to the optimal utility g^*. Proximity to the optimal solution thus relies entirely on our ability to avoid edge effects. We note that this technique of indirectly satisfying inequality (8) through the inequality (12) is quite novel, as we cannot achieve "super-fast" delay tradeoffs by working directly with the inequality (8).

To design such a control policy, we use a novel combination of *stochastic optimal Lyapunov scheduling* (developed in [1] [2] [3]), together with the concept of *buffer partitioning* from [4]. In particular, for a particular threshold parameter $Q > 0$ (to be determined later), we consider a policy that tends to decrease the queue backlog $U^{(c)}_n(t)$ when this backlog is greater than or equal to Q, and tends to *increase* queue backlog when $U^{(c)}_n(t) < Q$. This ensures stability while

also keeping backlog sufficiently far from the edge region. However, choosing a large value of Q will also directly increase average queue backlogs within the network.

A. Lyapunov Functions and Virtual Queues

Let $\underline{U}(t) = (U_n^{(c)}(t))$ represent the matrix of current queue backlogs. For given parameters $Q > 0$, $\omega > 0$ (to be determined later), we define the following *bi-modal Lyapunov function*:

$$L(\underline{U}) \triangleq \sum_{n \neq c} \left[e^{\omega(Q - U_n^{(c)}(t))} + e^{\omega(U_n^{(c)}(t) - Q)} - 2 \right] \quad (13)$$

This Lyapunov function achieves its minimum value of $L(\underline{U}) = 0$ when $U_n^{(c)} = Q$ for all $n \neq c$, and increases exponentially when the backlog in any queue deviates from the Q threshold either to the right or to the left. Minimizing the drift of this Lyapunov function from one timeslot to the next thus tends to maintain all queue backlogs near the Q threshold, and we shall find that the resulting edge probabilities decay exponentially in Q.

Next, we must ensure that the constraints (12) are satisfied. To this end. we use the concept of a *virtual queue* developed in [3]. For each active session $(n, c) \in \mathbb{M}$, we define a virtual queue $Z_{nc}(t)$ with a dynamic update equation as follows:

$$Z_{nc}(t+1) = \max[Z_{nc}(t) - \sum_b \mu_{nb}^{(c)}(t), 0] + \gamma_{nc}(t) + \sum_a \mu_{an}^{(c)}(t) \quad (14)$$

where $\mu_{ab}^{(c)}(t)$ and $\gamma_{nc}(t)$ are control decision variables used by the network controller. Note that the above update equation can be viewed as the dynamic equation of a discrete time queue with input process $\gamma_{nc}(t) + \sum_a \mu_{an}^{(c)}(t)$ and server process $\sum_b \mu_{nb}^{(c)}(t)$. Now define time averages $\overline{\gamma}_{nc}(t)$, $\overline{\mu}_{nc}^{(c)}(t)$ as follows:

$$\overline{\gamma}_{nc}(t) \triangleq \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{\gamma_{nc}(\tau)\}$$

$$\overline{\mu}_{ab}^{(c)}(t) \triangleq \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\left\{\mu_{ab}^{(c)}(\tau)\right\}$$

Further define values μ_{max}^{in} and μ_{max}^{out} as the maximum endogenous arrivals that can enter any given node during a single timeslot, and the maximum bits that can transmitted out of a given node during a single timeslot, respectively, considering all possible channel states and resource allocations. Specifically:

$$\mu_{max}^{in} \triangleq \sup_{n, \underline{S}, \underline{\mu} \in \Omega_{\underline{S}}} \sum_a \mu_{an} \ , \ \mu_{max}^{out} \triangleq \sup_{n, \underline{S}, \underline{\mu} \in \Omega_{\underline{S}}} \sum_b \mu_{nb}$$

Such values exist and are finite because the set of feasible rate matrices $\Omega_{\underline{S}}$ is compact for each of the finite number of channel states $\underline{S}$.

Lemma 1: If the network controller stabilizes all virtual queues $Z_{nc}(t)$, then:

$$\liminf_{t \to \infty} \left[\sum_b \overline{\mu}_{nb}^{(c)}(t) - \overline{\gamma}_{nc}(t) - \sum_a \overline{\mu}_{an}^{(c)}(t) \right] \geq 0 \quad (15)$$

Proof: The proof follows directly from the fact that if a queue with a bounded transmission rate is stable, then the $\liminf$ of the difference between the time average transmission rate and arrival rate is non-negative [25] [35]. □

The above lemma ensures that stabilizing all virtual queues yields inequalities similar to (12).

B. The Tradeoff-Optimal Control Policy

Assume that exogenous arrivals to any node are bounded by a value A_{max} on every timeslot, so that:

$$\sum_c A_{nc}(t) \leq A_{max} \quad \text{for all } n \in \{1, \ldots, N\}$$

For a given threshold Q (to be determined later), define indicator functions $1_{nc}^L(t)$ and $1_{nc}^R(t)$ for all (n, c) as follows:

$$1_{nc}^L(t) \triangleq \begin{cases} 1 & \text{if } U_n^{(c)}(t) < Q \\ 0 & \text{if } U_n^{(c)}(t) \geq Q \end{cases}$$

and $1_{nc}^R(t) = 1 - 1_{nc}^L(t)$. That is, $1_{nc}^L(t)$ and $1_{nc}^R(t)$ take the value 1 if and only if the corresponding queue backlog is to the "Left" and "Right" of the Q threshold, respectively. Let R_{max} be any value greater than or equal to A_{max}, and recall that $L_{nc}(t)$ is the amount of data in storage reservoir (n, c). We have the following control algorithm, defined in terms of positive constants ω, Q, V (to be determined later).

Utility-Delay Optimal Algorithm (UDOA): Initialize all virtual queues so that $Z_{nc}(0) = 0$ for all (n, c). The control policy is decoupled into the following policies for flow control, routing, and resource allocation, implemented every timeslot:

- *Flow Control:* The flow controller at each node n observes $U_n^{(c)}(t)$ and $Z_{nc}(t)$ and makes the following decisions for each commodity $c \in \{1, \ldots, N\}$:
  1) If $U_n^{(c)}(t) \geq Q$, then choose $R_{nc}(t) = 0$. Next, list all remaining (n, c) streams at node n in order of increasing $U_n^{(c)}(t)$, and sequentially assign $R_{nc}(t)$ to be as large as possible for the commodities c with the smallest values of $U_n^{(c)}(t)$ (noting that $R_{nc}(t) \leq A_{nc}(t) + L_{nc}(t)$), subject to the constraint $\sum_c R_{nc}(t) \leq R_{max}$.
  2) Choose $\gamma_{nc}(t) = \gamma_{nc}$, where γ_{nc} solves:

$$\begin{aligned} \text{Maximize:} \quad & V g_{nc}(\gamma_{nc}) - 2 Z_{nc}(t) \gamma_{nc} \\ \text{Subject to:} \quad & 0 \leq \gamma_{nc} \leq R_{max} \end{aligned}$$

  The virtual queues $Z_{nc}(t)$ are then updated according to (14), using the $\gamma_{nc}(t)$ values computed above and the $\mu_{an}^{(c)}(t)$, $\mu_{nb}^{(c)}(t)$ values computed by the following routing and resource allocation algorithms.
- *Routing:* The controller at each node n observes the queue backlogs of its neighboring nodes, and for each neighbor node b and each commodity c such that $(n, b) \in \mathbb{L}_c$, it computes $W_{nb}^c(t)$, defined as follows:

$$\begin{aligned} W_{nb}^c(t) \triangleq & \left[1_{nc}^R(t) e^{\omega(U_n^{(c)}(t) - Q)} - 1_{bc}^R(t) e^{\omega(U_b^{(c)}(t) - Q)} \right] \\ & - \left[1_{nc}^L(t) e^{\omega(Q - U_n^{(c)}(t))} - 1_{bc}^L(t) e^{\omega(Q - U_b^{(c)}(t))} \right] \\ & + \frac{2}{\omega} \left[Z_{nc}(t) - Z_{bc}(t) \right] \end{aligned}$$

The *optimal commodity* $c^*_{nb}(t)$ is then chosen as the commodity c^* that maximizes $W^c_{nb}(t)$ over all $c \in \{1, \ldots, N\}$ such that $(n, b) \in \mathbb{L}_c$ (ties are broken arbitrarily). Define $W^*_{nb}(t) \triangleq \max[W^{c^*}_{nb}(t), 0]$. Routing variables are then chosen as follows:

$$\mu^{(c)}_{nb}(t) = \begin{cases} 0 & \text{if } W^*_{nb}(t) = 0 \text{ or } c \neq c^*_{nb}(t) \\ \mu_{nb}(t) & \text{otherwise} \end{cases}$$

where the transmission rates $\mu_{nb}(t)$ are computed by the resource allocation algorithm below.

- *Resource Allocation:* Every timeslot t the network controllers observe the current channel state matrix $\underline{S}(t)$ and allocate transmission rates $(\mu_{ab}(t)) = (\mu_{ab})$, where (μ_{ab}) solves the following optimization problem:

$$\begin{array}{ll} \text{Maximize:} & \sum_{ab} W^*_{ab}(t)\mu_{ab} \\ \text{Subject to:} & (\mu_{ab}) \in \Omega_{\underline{S}(t)} \end{array}$$

where the $W^*_{ab}(t)$ weights are obtained from the above routing algorithm.

The flow control, routing, and resource allocation layers are decoupled, but the algorithms effect each other through key parameters that are passed between layers. The flow control algorithm can be implemented separately at each node using only queue length information of that node. Note that the computation of $\gamma_{nc}(t)$ for each (n, c) is a simple convex optimization of one variable, and can easily be solved in real time for any concave utility function.

The routing algorithm can also be implemented in a distributed manner provided that nodes are aware of the backlog levels of their neighbors. It is interesting to note that if a given node has backlog that is below the Q threshold, the weights are impacted negatively which tends to reduce the amount of data transmitted from this node and increase the amount of data transmitted *to* this node. The opposite occurs in the case when backlog is above the Q threshold.

The resource allocation policy is the most complex part of the UDOA algorithm, but can be implemented in a distributed fashion in the case when channels can be decomposed into independent sets, as described by equation (1), or can be approximated using the distributed techniques of [26] [1]. Note that the resource allocation maximizes a weighted sum of instantaneous transmission rates, as in [26] [21]. However, this is complemented by completely new flow control and routing algorithms that use weight terms that turn ON or OFF depending on a buffer threshold. Below we show that the algorithm yields a "super-fast" utility-delay tradeoff curve.

C. Algorithm Performance

Let (r^*_{nc}) represent an optimally fair operating point that solves (6). We assume that all active input streams have rates that dominate the optimally fair operating point, and that $r^*_{nc} > 0$ for these streams. Specifically, we define ϵ_{max} as the largest value of ϵ such that there is an optimally fair rate point (r^*_{nc}) that satisfies: $\epsilon \leq r^*_{nc} \leq \lambda_{nc} - \epsilon$ for all active sessions $(n, c) \in \mathbb{M}$ (see Fig. 1). To analyze the UDOA algorithm, it is useful to define δ_{max} as the largest possible change in the total queue backlog at any node during a timeslot:

$$\delta_{max} \triangleq \max[R_{max} + \mu^{in}_{max}, \mu^{out}_{max}]$$

Define time averages $\overline{U}^{(c)}_n(t)$ and $\overline{r}_{nc}(t)$ as follows:

$$\begin{aligned} \overline{U}^{(c)}_n(t) &\triangleq \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\left\{U^{(c)}_n(\tau)\right\} \\ \overline{r}_{nc}(t) &\triangleq \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\left\{r_{nc}(\tau)\right\} \end{aligned}$$

Theorem 2: (UDOA Performance) Fix parameters V, ϵ such that $V > 0$ and $0 < \epsilon \leq \epsilon_{max}$, and choose any positive value ω that satisfies:

$$\omega\delta_{max}e^{\omega\delta_{max}} \leq \epsilon/\delta_{max} \tag{16}$$

Further define:[3]

$$Q \triangleq \frac{2}{\omega}\log(V) \tag{17}$$

Then the UDOA algorithm stabilizes all actual and virtual queues of the system, and yields:

$$\begin{aligned} \limsup_{t\to\infty} \frac{1}{N^2}\sum_{nc} \overline{U}^{(c)}_n(t) &\leq O\left(\log(V)\right) \\ \liminf_{t\to\infty} \sum_{nc} g_{nc}\left(\overline{r}_{nc}(t)\right) &\geq g^* - O(1/V) \end{aligned}$$

Proof: The proof is developed in Appendix A, where explicit utility and delay bounds are derived. □

Because V is an arbitrary control parameter, it can be made as large as desired, pushing total system utility arbitrarily close to the optimal utility g^* with a corresponding logarithmic increase in average congestion (and hence, average delay). Note that this performance holds *regardless of the reservoir buffer size of the flow controllers*. Thus, a non-zero reservoir buffer does not contribute to achieving these super-fast tradeoffs. However, large reservoir buffers can be used in practice to preserve data from individual streams and maintain FIFO packet admission.

Note that if the ω parameter is chosen as follows:

$$\omega \triangleq \frac{\epsilon}{\delta^2_{max}}e^{-\epsilon/\delta_{max}}$$

then the inequality (16) is necessarily satisfied. This follows directly from the fact that for any positive value c, the inequality $xe^x \leq c$ is satisfied by the variable $x = ce^{-c}$.

V. Optimality of the Logarithmic Tradeoff

Here we show that it is not possible to improve upon the $O(\log(V))$ delay characteristic, and hence our scheduling algorithm captures the tightest tradeoff. To simplify analysis, we concentrate on the special case of a *single hop network* with L data links and L data streams with rates $(\lambda_1, \ldots, \lambda_L)$, where data from stream $i \in \{1, \ldots, L\}$ is stored in a distinct queue for transmission over link i. We further assume the network has the following characteristics:

[3] All $\log(\cdot)$ functions in this paper denote the natural logarithm.

1) Flow control reservoirs have zero buffer space, so that admission/rejection decisions are made immediately upon packet arrival.[4]
2) Arrivals are i.i.d. over timeslots and independent of channel states, and there exists a probability $p > 0$ that no new packets arrive from *any* data stream.
3) Channel states are i.i.d. over timeslots, and there exists a rate θ and a probability q such that: For each link i, with probability at least q a channel state arises that would allow link i to transmit with rate at least θ (if all resources were to be completely devoted to link i).
4) If it is possible to transmit with rates $(\mu_1(t), \ldots, \mu_L(t))$ during slot t, then it is also possible to transmit with rates $(\tilde{\mu}_1(t), \ldots, \tilde{\mu}_L(t))$ during slot t, where $\tilde{\mu}_i(t) \leq \mu_i(t)$ for all $i \in \{1, \ldots, L\}$.

Assume that utility functions $g_i(r)$ are differentiable, strictly increasing, and concave, and that the input rate vector $\vec{\lambda}$ is finite and outside of the capacity region. Let r_{max} be the maximum possible transmission rate, and define $\beta \triangleq \min_{i\in\{1,\ldots,N\}} \frac{dg_i(r_{max})}{dr}$ to be the minimum possible slope of any utility function over the rate region of interest. Note that our assumptions imply that $\beta > 0$. Further, we restrict attention to the class of scheduling policies that are ergodic with well defined steady state averages.

Theorem 3: If a control policy of the type described above yields a total utility that differs from the optimal utility by no more than $1/V$, then average congestion must be greater than or equal to $\Omega(\log(V))$.

Proof: Consider an ergodic control policy that yields total throughput within $1/V$ of the optimal throughput $\sum_i g_i(r_i^*)$, where $(r_1^*, \ldots, r_L^*)$ solves (6). Let $(\overline{r}_1, \ldots, \overline{r}_L)$ represent the throughput vector achieved by this policy. Thus:

$$1/V \geq \sum_{i=1}^{L} g_i(r_i^*) - \sum_{i=1}^{L} g_i(\overline{r}_i) \tag{18}$$

Note that $(\overline{r}_1, \ldots, \overline{r}_L) \in \Lambda$, and $\overline{r}_i \leq \lambda_i$ for all i. Because $\vec{\lambda} \notin \Lambda$, there must be a link $m \in \{1, \ldots, L\}$ such that $\lambda_m > \overline{r}_m$. Define $\tilde{\theta} = \min[\theta, (\lambda_m - \overline{r}_m)]$.

Let $\overline{U}$ denote the total average bit occupancy in the system. Define $T = 2\lceil 2\overline{U}/\tilde{\theta} + L\rceil$. Let $U_1(t), \ldots, U_L(t)$ represent the queue backlogs in a particular slot t. We say that a T slot interval beginning at time t is an *edge interval* if the following sequence of events occurs:

- The total queue backlog $U_1(t)+\ldots+U_L(t)$ in the system at time t is less than or equal to $2\overline{U}$.
- There are no new arrivals to any queue of the system during the T slot interval.
- For the first $\lceil U_1(t)/\tilde{\theta}\rceil$ slots, channel states arise that would allow link 1 to transmit at rate at least θ. For the next $\lceil U_2(t)/\tilde{\theta}\rceil$ slots, channel states arise that would allow link 2 to transmit at rate at least θ, and so forth, so that all links sequentially would be able to transmit their full starting load, taking up a total of $\lceil U_1(t)/\tilde{\theta}\rceil + \ldots + \lceil U_L(t)/\tilde{\theta}\rceil$ timeslots. This is called the *first phase* of the interval. For all remaining timeslots up to T, channel states arise that would allow the special link m to transmit at rate at least θ.

[4]The same result can be proven when all flow control reservoirs have finite buffers, but the result is not true in the case when reservoirs have infinite buffers.

Assuming the system is in steady state at the beginning of an edge interval, by the Markov inequality for non-negative random variables we have:

$$Pr[\textstyle\sum_i U_i(t) \leq 2\overline{U}] \geq 1/2$$

Let δ represent the steady state probability of a particular set of T slots being an edge interval. We thus have:

$$\delta \geq \frac{1}{2}(pq)^T \geq \frac{1}{2}(pq)^{(4\overline{U}/\tilde{\theta}+2L+2)} \tag{19}$$

Note that the number of timeslots in the first phase of an edge interval satisfies:

$$\begin{aligned}\sum_{i=1}^{L}\lceil U_i(t)/\tilde{\theta}\rceil &\leq \sum_{i=1}^{L}(U_i(t)/\tilde{\theta}+1)\\ &\leq 2\overline{U}/\tilde{\theta} + L\\ &\leq T/2\end{aligned}$$

Hence, this phase takes up at most half of the slots of an edge interval. Further, if the controller knew in advance that this was an edge interval, it could clearly schedule so that all initial backlog is cleared during this first phase. In particular, it is possible to make a sequence of transmission decisions that yields a per-slot empirical average transmission rate during the first $T/2$ slots that is exactly equal to the per-slot empirical average throughput of the actual control policy over the full T slots.

Consider now an alternate transmission strategy that runs on a "parallel" system with the same channel states: The timeline $t \in \{0, 1, 2, \ldots\}$ is partitioned into successive disjoint intervals of T slots. If the current interval is an edge interval, during the first phase we transmit to yield a per-slot empirical average transmission rate exactly equal to the per-slot empirical average throughput attained by the actual control policy over the full T slots. During the second phase, we remain idle until slot $T/2$, after which we transmit only over link m, with an exact transmission rate of $\tilde{\theta}$ until the T slot interval expires. If the current interval is *not* an edge interval, each link transmits with a rate exactly equal to the amount of bits delivered over the link on that timeslot by the original policy. Note that this alternate transmission strategy is *non-causal* as it requires knowledge of future events to make the transmission decisions. However, the capacity region of the system contains all long term average throughput vectors achieved by either causal or non-causal policies [1] [26], and hence the resulting time average transmission rate vector $(\hat{r}_1, \ldots, \hat{r}_L)$ achieved by this alternate policy is within the capacity region Λ.

Further note that:

$$\hat{r}_i = \overline{r}_i \quad \text{for all } i \neq m \tag{20}$$

$$\hat{r}_m = \overline{r}_m + \delta\tilde{\theta}/2 \tag{21}$$

where the second equality follows because, on any edge interval, the alternate policy yields a total average transmission rate on link m that is exactly $\tilde{\theta}/2$ beyond the link m average throughput of the control policy over this interval. From (20) it follows that $\hat{r}_i \leq \lambda_i$ for all $i \neq m$. From (21) it follows that

$\hat{r}_m \leq \overline{r}_m + \delta \min[\theta, (\lambda_m - \overline{r}_m)] \leq \overline{r}_m + (\lambda_m - \overline{r}_m)$ and so $\hat{r}_m \leq \lambda_m$.

It follows that $(\hat{r}_1, \ldots, \hat{r}_m)$ satisfies the constraints of the optimization problem (6), and so its utility is less than or equal to the optimal utility. Hence, from (18):

$$\begin{aligned} 1/V &\geq \sum_{i=1}^{L} g_i(r_i^*) - \sum_{i=1}^{L} g_i(\overline{r}_i) \\ &\geq \sum_{i=1}^{L} g_i(\hat{r}_i) - \sum_{i=1}^{L} g_i(\overline{r}_i) \\ &= g_m(\overline{r}_m + \delta\tilde{\theta}/2) - g_m(\overline{r}_m) \end{aligned}$$

where the last line follows by (20) and (21). Using the definition of β, we have:

$$1/V \geq \beta\delta\tilde{\theta}/2$$

Using the inequality (19), we have:

$$\frac{1}{V} \geq \frac{1}{4}\beta\tilde{\theta}(pq)^{(4\overline{U}/\tilde{\theta}+2L+2)} \tag{22}$$

Taking the logarithm of both sides and shifting terms yields:

$$\overline{U} \geq \frac{\tilde{\theta}}{4}\left(\frac{\log(V\beta\tilde{\theta}/4)}{\log(1/(pq))} - 2L - 2\right)$$

Hence, average backlog grows at least logarithmically in V. □

VI. Extensions and Open Questions

These super-fast convergence results can be extended to treat networks with average cost constraints, such as the power constrained networks described in [3], provided that input rates exceed the corresponding network capacity. However, such "super-fast" logarithmic delay tradeoffs cannot be obtained in related problems of minimizing average cost. Indeed, such problems are fundamentally different, and lead to square-root law tradeoffs [4] [32]. One reason for this difference is that the average cost problem can be viewed in terms of optimizing an expectation, rather than in terms of optimizing a concave function of an expectation. Another difference is that the problem we have treated here assumes input rates are outside of the capacity region, while problems of minimizing average cost assume input rates are strictly interior to the capacity region and thus can be stabilized.

A. Decay Exponents

An alternate interpretation of the $[O(1/V); O(\log(V))]$ utility-delay result of Theorem 2 is that, under the UDOA control algorithm, linearly increasing network congestion leads to exponentially fast convergence of network utility. Explicit utility and congestion bounds that describe the rate of this exponential convergence are computed in Appendix A. This leads to the following corollary:

Corollary 1: For any $\omega > 0$ that satisfies $\omega\delta_{max}e^{\omega\delta_{max}} \leq \epsilon/\delta_{max}$, and for any $V > 0$ and Q as defined in (17), the UDOA algorithm implemented with V, ω, Q yields:

$$\liminf_{t\to\infty} \sum_{nc} g_{nc}(\overline{r}_{nc}(t)) \geq g^* - O(e^{-\frac{\omega}{3}\overline{U}})$$

where

$$\overline{U} \triangleq \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \frac{1}{N^2}\sum_{nc} \mathbb{E}\left\{U_n^{(c)}(\tau)\right\}$$

and where $\overline{U}$ grows logarithmically in the V parameter.

Proof: See Appendix B. □

B. Open Questions

We have proven that the $[O(1/V); O(\log(V))]$ utility-delay tradeoff is optimal. Hence, it is possible to control a general network to achieve utility that converges exponentially fast to the optimal utility, with a corresponding linear delay increase. Conversely, super-exponential convergence is impossible. However, an interesting open question is to find the *maximum decay exponent*. Indeed, we can define η as the maximum value such that it is possible to push utility within $O(e^{-\eta\overline{U}})$ of the optimal utility. Our achievability result in Corollary 1 places a lower bound on η, and our analysis of the necessity of logarithmic delay, leading to inequality (22), places an upper bound on η. Specifically, we have proven that:

$$\frac{\omega^*}{3} \leq \eta \leq \frac{4\log(1/(pq))}{\tilde{\theta}}$$

where ω^* is defined as the maximum value of ω that satisfies $\omega\delta_{max}e^{\omega\delta_{max}} \leq \epsilon_{max}/\delta_{max}$ (and in particular, $\omega^* \geq (\epsilon_{max}/\delta_{max}^2)e^{-\epsilon_{max}/\delta_{max}}$). Note that, strictly speaking, the upper bound was derived only for one-hop networks with the properties described in Section V, although the lower bound applies to any general multihop network with concave utility functions. It is remarkable that this lower bound can be expressed purely in terms of the ϵ_{max} and δ_{max} parameters.

VII. Simulation of a Wireless Downlink

For simplicity, here we simulate the simple 2-queue downlink example of [2] with a single server and two channels with randomly varying ON/OFF states. Each channel i is independently ON with probability q_i, where $(q_1, q_2) = (0.6, 0.5)$. As this is a single hop system, there are no routing decisions, and the network controller only decides which ON queue to serve (if any) during a timeslot. Assume packets arrive according to independent Bernoulli processes with arrival probabilities $(\lambda_1, \lambda_2) = (1.0, 0.5)$, and that there are no storage reservoirs. We implement UDOA with $\epsilon = 0.2, R_{max} = 2$, and plot average congestion versus V together with the resulting system throughput (Figs. 2, 3). Note that the flow controller in this example reduces to allowing a new arrival if and only if $U_i(t) < Q$, and hence $U_i(t) \leq Q + 1$ for all t. The figure illustrates simulation results when the V parameter is varied between 5 and $10,000$, together with the upper bound on congestion from (40). Each data point represents a simulation over 5 million timeslots. We use utility functions $g_1(r) = \log(1 + r)$, $g_2(r) = 1.28\log(1 + r)$, which yields an optimal operating point of $(\overline{r}_1, \overline{r}_2) = (0.23, 0.57)$. From the figure, it is clear that the resulting throughput approaches this optimal operating point, while average congestion increases logarithmically in V (the plot is linear on a log scale).

Note that the UDOA policy may choose to serve *no packets* even when both queues are full and both channels are ON. This

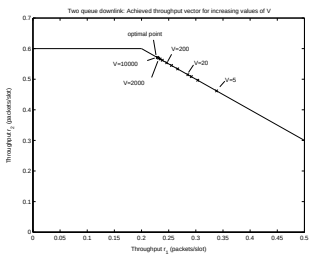

Fig. 2. Illustration of the achieved throughput (plotted on the capacity region) for different values of V.

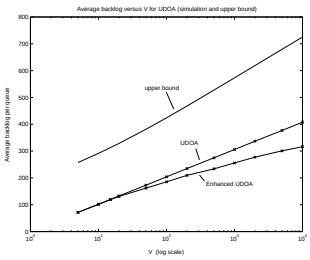

Fig. 3. Average number of packets versus V, illustrating growth that is logarithimc in V.

policy can be modified via an "enhanced" strategy that maintains an *actual queue state* $\tilde{U}_i(t)$ in addition to the original $U_i(t)$ queue state. The actual queue state allows packets to be scheduled in such scenarios, but otherwise all flow control and scheduling decisions are made according to the $U_i(t)$ backlog. It is not difficult to see that this modified policy yields the same throughput as the original, while ensuring $\tilde{U}_i(t) \leq U_i(t)$ for all t (in cases when strict inequality holds, the $U_i(t)$ queue is viewed as containing *phantom packets*). The congestion of this enhanced policy is also shown in Fig. 3.

The value of Q in (17) was chosen to analytically prove that edge effects arise with very low probability. Our analysis was quite conservative, and in the above experiments it was observed that, after the third timeslot, no queues ever emptied during the course of the 5 million iterations. Experimentally, it was found that the algorithm can be implemented with a value of Q that is reduced by a factor of 40, yielding average queue backlog that is also roughly reduced by a factor of 40, with no significant degradation in throughput in any of the experiments (results omitted for brevity). This suggests that even further (constant factor) improvements are possible by treating Q as a dynamic control variable that is adjusted in reaction to the observed edge probability.

VIII. Conclusions

This work presents a theory of utility and delay tradeoffs for general data networks with stochastic channel conditions and randomly arriving traffic. A novel control technique was developed and shown to push total system utility arbitrarily close to the optimal operating point, with a corresponding logarithmic tradeoff in average end-to-end network delay. The algorithm is decoupled into separate strategies for flow control, routing, and resource allocation, and can be implemented without requiring knowledge of traffic rates or channel statistics. Further, we proved that no other algorithm can improve upon the logarithmic tradeoff curve. This work establishes a new and important relationship between utility and delay, and introduces fundamentally new techniques for performance optimal scheduling. These techniques can likely be applied in other areas of stochastic optimization and control to yield simple solutions that offer significant performance improvements.

Appendix A — Performance Analysis

Here we prove the performance bounds expressed in Theorem 2 for the UDOA control algorithm. We first review a central result from [1] [3] [2] concerning performance optimal Lyapunov analysis, presented here in a modified form. Consider any queueing network with queue backlogs expressed as a vector $\vec{X}(t) = (X_1(t), \ldots, X_M(t))$. Let $\vec{R}(t) = (R_1(t), \ldots, R_M(t))$ represent an associated control process confined to some compact control space. Let $g(\vec{r})$ be any non-negative, concave utility function, and let $g(\vec{r}^*)$ represent a target utility value for the time average of the $\vec{R}(t)$ process. Define g_{max} as the maximum of $g(\vec{R}(t))$ over the control space. Let $\Psi(\vec{X})$ be a non-negative function of the queue backlog vector. We call $\Psi(\vec{X})$ a *Lyapunov function*, and define the *conditional Lyapunov drift* $\Delta(\vec{X}(t))$ as follows:

$$\Delta(\vec{X}(t)) \triangleq \mathbb{E}\left\{\Psi(\vec{X}(t+1)) - \Psi(\vec{X}(t)) \mid \vec{X}(t)\right\}$$

Lemma 2: (Lyapunov Drift with Performance Optimization) If there exist positive values B, ϵ, V and a non-negative function $f(\vec{X})$ such that every timeslot t the conditional Lyapunov drift satisfies:

$$\Delta(\vec{X}(t)) - V\mathbb{E}\left\{g(\vec{R}(t)) \mid \vec{X}(t)\right\} \leq B - \epsilon f(\vec{X}(t)) - Vg(\vec{r}^*)$$

then we have:

$$\begin{aligned} \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\left\{f(\vec{X}(\tau))\right\} &\leq \frac{B + Vg_{max}}{\epsilon} \\ \liminf_{t\to\infty} g(\overline{r}_1(t), \ldots, \overline{r}_M(t)) &\geq g(\vec{r}^*) - B/V \end{aligned}$$

where

$$\overline{r}_m(t) \triangleq \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\left\{R_m(\tau)\right\} \text{ for } m \in \{1, \ldots, M\}$$

Proof: The proof is almost identical to the proofs of similar statements given in [1] [3] [2], and is omitted for brevity. □

If V is a free control variable, then the lemma shows that total utility can be pushed arbitrarily close to the optimal utility value, with a corresponding linear increase in the time average of $f(\vec{X}(t))$. Below we construct a Lyapunov function and show that the UDOA control law yields a drift similar to the form given in the lemma above.

A. Computing the Drift

Define $\underline{X}(t) \triangleq [\underline{U}(t); \underline{Z}(t)]$ as the collective state of the actual and virtual queues of the system. Consider the Lyapunov function $L(\underline{U})$ defined in (13), and let $\Delta_L(\underline{X}(t))$ represent the conditional drift. Note that this is a function of the full system state because control decisions that effect the drift can, in general, depend on both $\underline{U}(t)$ and $\underline{Z}(t)$. To compute $\Delta_L(\underline{X}(t))$, for notational convenience we define the following set of variables $\delta_{nc}(t)$:

$$\delta_{nc}(t) \triangleq \sum_b \mu_{nb}^{(c)}(t) - \sum_a \mu_{an}^{(c)}(t) - R_{nc}(t) \qquad (23)$$

Note by definition that $|\delta_{nc}(t)| \leq \delta_{max}$.

Lemma 3: For any $\omega > 0$, we have:

$$\sum_{nc} e^{\omega(U_n^{(c)}(t+1) - Q)} \leq N^2 e^{\omega(2\delta_{max} - Q)} + \sum_{nc} e^{\omega(U_n^{(c)}(t) - Q)} \left[1 - \omega\delta_{nc}(t) + \frac{\omega^2\delta_{max}^2}{2} e^{\omega\delta_{max}}\right]$$

Likewise:

$$\sum_{nc} e^{-\omega(Q-U_n^{(c)}(t+1))} \leq \sum_{nc} e^{\omega(Q-U_n^{(c)}(t))}\left[1+\omega\delta_{nc}(t)+\frac{\omega^2\delta_{max}^2}{2}e^{\omega\delta_{max}}\right]$$

Proof: Define $\tilde{\mu}_{nb}^{(c)}(t)$ as the amount of commodity c data actually delivered over link (n,b) during timeslot t, and note that $\tilde{\mu}_{nb}^{(c)}(t) \leq \mu_{nb}^{(c)}(t)$, with equality whenever $U_n^{(c)}(t) \geq \mu_{max}^{out}$. The dynamic queueing inequality (4) can thus be re-written:

$$U_n^{(c)}(t+1) \leq U_n^{(c)}(t) - \sum_b \tilde{\mu}_{nb}^{(c)}(t) + \sum_a \mu_{an}^{(c)}(t) + R_{nc}(t)$$

It follows that:

$$U_n^{(c)}(t+1) \leq \begin{cases} \mu_{max}^{out}+\mu_{max}^{in}+R_{max} & \text{if } U_n^{(c)}(t) < \mu_{max}^{out} \\ U_n^{(c)}(t)-\delta_{nc}(t) & \text{if } U_n^{(c)}(t) \geq \mu_{max}^{out} \end{cases}$$

Noting that $\mu_{max}^{out}+\mu_{max}^{in}+R_{max} \leq 2\delta_{max}$, we have:

$$\begin{aligned} e^{\omega U_n^{(c)}(t+1)} &\leq e^{2\omega\delta_{max}} + e^{\omega U_n^{(c)}(t)}e^{-\omega\delta_{nc}(t)} \\ &\leq e^{2\omega\delta_{max}} + e^{\omega U_n^{(c)}(t)}\left[1-\omega\delta_{nc}(t)+\frac{\omega^2\delta_{max}^2}{2}e^{\omega\delta_{max}}\right] \end{aligned}$$

where the final inequality follows by the Taylor expansion of $e^{-\omega\delta}$, noting that $|\delta_{nc}(t)| \leq \delta_{max}$. Summing the above inequality over all nodes n and c and multiplying by $e^{-\omega Q}$ proves the first part of the lemma. The second part is proved similarly. □

Lemma 4: If ω satisfies (16), then:

$$\begin{aligned} \Delta_L(\underline{X}(t)) \leq\ & N^2e^{\omega(2\delta_{max}-Q)} + N^2\omega(2\delta_{max}+\epsilon) \\ & -\omega\sum_{nc} 1_{nc}^R(t)e^{\omega(U_n^{(c)}(t)-Q)}\left[\mathbb{E}\left\{\delta_{nc}(t)\mid \underline{X}(t)\right\}-\frac{\epsilon}{2}\right] \\ & -\omega\sum_{nc} 1_{nc}^L(t)e^{\omega(Q-U_n^{(c)}(t))}\left[\mathbb{E}\left\{-\delta_{nc}(t)\mid \underline{X}(t)\right\}-\frac{\epsilon}{2}\right] \end{aligned}$$

Proof: Note that if ω satisfies (16), then:

$$\frac{\omega^2\delta_{max}^2}{2}e^{\omega\delta_{max}} \leq \frac{\omega\epsilon}{2}$$

Substituting this inequality into the expressions of Lemma 3, shifting and summing terms, and taking expectations yields:

$$\begin{aligned} \Delta_L(\underline{X}(t)) \leq\ & N^2e^{\omega(2\delta_{max}-Q)} \\ & -\omega\sum_{nc} e^{\omega(U_n^{(c)}(t)-Q)}\left[\mathbb{E}\left\{\delta_{nc}(t)\mid \underline{X}(t)\right\}-\frac{\epsilon}{2}\right] \\ & -\omega\sum_{nc} e^{\omega(Q-U_n^{(c)}(t))}\left[\mathbb{E}\left\{-\delta_{nc}(t)\mid \underline{X}(t)\right\}-\frac{\epsilon}{2}\right] \quad (24) \end{aligned}$$

Now recall that $1_{nc}^L(t) = 0$ whenever $U_n^{(c)}(t) \geq Q$, and hence for any positive or negative value C we have:

$$\begin{aligned} e^{\omega(U_n^{(c)}(t)-Q)}C &= (1_{nc}^L(t)+1_{nc}^R(t))e^{\omega(U_n^{(c)}(t)-Q)}C \\ &\leq |C| + 1_{nc}^R(t)e^{\omega(U_n^{(c)}(t)-Q)}C \quad (25) \end{aligned}$$

Likewise:

$$e^{\omega(Q-U_n^{(c)}(t))}C \leq |C| + 1_{nc}^L(t)e^{\omega(Q-U_n^{(c)}(t))}C \quad (26)$$

Using $C = -\omega(\delta_{nc}(t)-\epsilon/2)$ in (25) and $C = -\omega(-\delta_{nc}(t)-\epsilon/2)$ in (26) and plugging the resulting inequalities into (24) yields the result. □

Lemma 4 establishes the drift of the Lyapunov function $L(\underline{U}(t))$ associated with the actual queues of the system. Now let $\underline{Z}(t) = (Z_{nc}(t))$ represent the matrix of virtual queue processes, and define an additional Lyapunov function $H(\underline{Z}(t))$ as follows:

$$H(\underline{Z}(t)) \triangleq \sum_{nc} Z_{nc}^2(t)$$

Define $\Delta_H(\underline{X}(t))$ as the conditional drift of the Lyapunov function $H(\underline{Z}(t))$. Using the update equation (14) for the $Z_{nc}(t)$ queues together with a standard computation for the drift of a quadratic Lyapunov function, we have (see, for example, [21] [26] [2]):

$$\begin{aligned} \Delta_H(\underline{X}(t)) \leq\ & 4N^2\delta_{max}^2 \\ & -2\sum_{nc} Z_{nc}(t)\mathbb{E}\left\{\sum_b \mu_{nb}^{(c)}(t) - \sum_a \mu_{an}^{(c)}(t)\mid \underline{X}(t)\right\} \\ & +2\sum_{nc} Z_{nc}(t)\mathbb{E}\left\{\gamma_{nc}(t)\mid \underline{X}(t)\right\} \quad (27) \end{aligned}$$

B. Controlling the Drift

Define an aggregate Lyapunov function $\Psi(\underline{X}(t)) = L(\underline{U}(t))+H(\underline{Z}(t))$, and let $\Delta(\underline{X}(t))$ represent the conditional drift. Thus:

$$\begin{aligned} \Delta(\underline{X}(t)) - V\mathbb{E}\left\{\textstyle\sum_{nc} g_{nc}(\gamma_{nc}(t))\mid \underline{X}(t)\right\} &= \Delta_L(\underline{X}(t)) \\ &+\Delta_H(\underline{X}(t)) - V\mathbb{E}\left\{\textstyle\sum_{nc} g_{nc}(\gamma_{nc}(t))\mid \underline{X}(t)\right\} \end{aligned}$$

where we have subtracted the same term from both sides of the above equality. Using the bounds on $\Delta_L(\underline{X}(t))$ and $\Delta_H(\underline{X}(t))$ given in Lemma 4 and in (27), we have:

$$\begin{aligned} \Delta(\underline{X}(t)) - V\mathbb{E}\left\{\sum_{nc} g_{nc}(\gamma_{nc}(t))\mid \underline{X}(t)\right\} \leq\ & N^2B \\ & -\omega\sum_{nc} 1_{nc}^R(t)e^{\omega(U_n^{(c)}(t)-Q)}\left[\mathbb{E}\left\{\delta_{nc}(t)\mid \underline{X}(t)\right\}-\frac{\epsilon}{2}\right] \\ & -\omega\sum_{nc} 1_{nc}^L(t)e^{\omega(Q-U_n^{(c)}(t))}\left[\mathbb{E}\left\{-\delta_{nc}(t)\mid \underline{X}(t)\right\}-\frac{\epsilon}{2}\right] \\ & -2\sum_{nc} Z_{nc}(t)\mathbb{E}\left\{\sum_b \mu_{nb}^{(c)}(t) - \sum_a \mu_{an}^{(c)}(t)\mid \underline{X}(t)\right\} \\ & +2\sum_{nc} Z_{nc}(t)\mathbb{E}\left\{\gamma_{nc}(t)\mid \underline{X}(t)\right\} \\ & -V\mathbb{E}\left\{\sum_{nc} g_{nc}(\gamma_{nc}(t))\mid \underline{X}(t)\right\} \quad (28) \end{aligned}$$

where $B \triangleq e^{\omega(2\delta_{max}-Q)} + \omega(2\delta_{max}+\epsilon) + 4\delta_{max}^2$.

The UDOA algorithm is derived directly from the drift bound (28). Indeed, the flow control, routing, and resource allocation algorithms of UDOA *were constructed specifically to minimize the right hand side of (28) over all possible choices of the control variables* $\gamma_{nc}(t), \mu_{ab}^{(c)}(t), R_{nc}(t)$. We show this in more detail below by isolating the control variables corresponding to each layer of the algorithm.

Flow Control: Isolating the $\gamma_{nc}(t)$ variables that occur on the right hand side of (28), we have:

$$\mathbb{E}\left\{2Z_{nc}(t)\gamma_{nc}(t) - Vg_{nc}(\gamma_{nc}(t)) \mid \underline{X}(t)\right\}$$

The UDOA flow control algorithm minimizes these terms for each (n,c) by choosing $\gamma_{nc}(t)$ to maximize $Vg_{nc}(\gamma_{nc}) - 2Z_{nc}(t)\gamma_{nc}$ over $0 \leq \gamma_{nc} \leq R_{max}$.

Likewise, isolating the $R_{nc}(t)$ variables in (28) using the definition of $\delta_{nc}(t)$ in (23), we have:

$$\sum_{nc}\mathbb{E}\left\{R_{nc}(t) \mid \underline{X}(t)\right\}\omega\left[1^R_{nc}(t)e^{\omega(U_n^{(c)}(t)-Q)} - 1^L_{nc}(t)e^{\omega(Q-U_n^{(c)}(t))}\right]$$

The positive terms of the above sum occur only when $U_n^{(c)}(t) \geq Q$, and these terms are minimized by the UDOA policy of assigning $R_{nc}(t) = 0$. The remaining sum has all negative terms, and is minimized by choosing the remaining $R_{nc}(t)$ values according to the UDOA flow control policy. In particular, such $R_{nc}(t)$ decisions make the value of the above summation less than or equal to the value obtained by any algorithm that chooses $R_{nc}(t)$ by probabilistically admitting or rejecting all incoming data $A_{nc}(t)$ according to some specific probabilities. Actually, any flow control algorithm that yields $\sum_c R_{nc}(t)1^L_{nc}(t)e^{-\omega U_n^{(c)}(t)} \geq \sum_c A_{nc}(t)1^L_{nc}(t)e^{-\omega U_n^{(c)}(t)}$ for all t, n is sufficient for this purpose and could also be used.

Routing: Isolating the $\mu^{(c)}_{nb}(t)$ variables in the right hand side of (28), we have:

$$\sum_c \mathbb{E}\left\{\mu^{(c)}_{nb}(t) \mid \underline{X}(t)\right\}\Big[-2Z_{nc}(t) + 2Z_{bc}(t) - \omega 1^R_{nc}(t)e^{\omega(U_n^{(c)}(t)-Q)} + \omega 1^R_{bc}(t)e^{\omega(U_b^{(c)}(t)-Q)} + \omega 1^L_{nc}(t)e^{\omega(Q-U_n^{(c)}(t))} - \omega 1^L_{bc}(t)e^{\omega(Q-U_b^{(c)}(t))}\Big]$$

which is equal to:

$$-\omega\mathbb{E}\left\{\sum_{nbc}\mu^{(c)}_{nb}(t)W^{(c)}_{nb}(t) \mid \underline{X}(t)\right\}$$

where $W^{(c)}_{nb}(t)$ is as defined in the UDOA routing algorithm. The UDOA routing algorithm maximizes $\sum_{nbc}\mu^{(c)}_{nb}(t)W^{(c)}_{nb}(t)$ over all $\mu^{(c)}_{nb}(t)$ variables that satisfy $\sum_c \mu^{(c)}_{nb}(t) \leq \mu_{nb}(t)$ and $\mu^{(c)}_{nb}(t) = 0$ if $(n,b) \notin \mathbb{L}_c$. To see this, note that:

$$\begin{aligned}\sum_{nbc}\mu^{(c)}_{nb}(t)W^{(c)}_{nb}(t) &\leq \sum_{nbc}\mu^{(c)}_{nb}(t)\max_{\{m|(n,b)\in\mathbb{L}_m\}}[W^{(m)}_{nb}(t), 0] \\ &= \sum_{nbc}\mu^{(c)}_{nb}(t)W^*_{nb}(t) \\ &\leq \sum_{nb}\mu_{nb}(t)W^*_{nb}(t)\end{aligned}$$

and that the upper bound is achieved by setting $\mu^{(c)}_{nb}(t) = \mu_{nb}(t)$ for $c = c^*$, and $\mu^{(c)}_{nb}(t) = 0$ else.

Resource Allocation: The resulting contribution of the $\mu^{(c)}_{nb}(t)$ terms to the drift expression (28) is thus given by:

$$-\omega\mathbb{E}\left\{\textstyle\sum_{nb}\mu_{nb}(t)W^*_{nb}(t) \mid \underline{X}(t)\right\}$$

which is minimized by the UDOA algorithm over all possible choices of the transmission rate matrix subject to the current channel condition: $(\mu_{nb}(t)) \in \Omega_{\underline{S}(t)}$.

C. Computing Time Averages

As the UDOA algorithm chooses control variables $\gamma_{nc}(t), \mu^{(c)}_{nb}(t), R_{nc}(t)$ that minimize all but the final term on the right hand side of the drift expression (28), we can establish a simpler bound by plugging in decision variables $\gamma^*_{nc}(t), \mu^{*(c)}_{nb}(t), R^*_{nc}(t)$ corresponding to alternative control policies.

To this end, let (r^*_{nc}) represent the optimal solution of (6). Because $(r^*_{nc}) \in \Lambda$, we know by (5) of Theorem 1 that there must exist a stationary randomized routing and resource allocation policy that chooses variables $\mu^{*(c)}_{nb}(t)$ independently of queue backlog, and yields:

$$\mathbb{E}\left\{\sum_b \mu^{*(c)}_{nb}(t) - \sum_a \mu^{*(c)}_{ab}(t) \mid \underline{X}(t)\right\} = r^*_{nc} \quad (29)$$

Now consider the flow control policy that chooses:

$$\gamma^*_{nc}(t) = r^*_{nc} \text{ for all } t \text{ and all } (n,c) \quad (30)$$

Further, consider the flow control algorithm that makes independent probabilistic admission decisions $R^*_{nc}(t)$ every timeslot and for each $(n,c) \in \mathbb{M}$ as follows:

$$R_{nc}(t) = \begin{cases} A_{nc}(t) & \text{with probability } p^L_{nc} \text{ if } U^{(c)}_n(t) < Q \\ A_{nc}(t) & \text{with probability } p^R_{nc} \text{ if } U^{(c)}_n(t) \geq Q \\ 0 & \text{otherwise} \end{cases}$$

where $p^L_{nc} = (r^*_{nc} + \epsilon)/\lambda_{nc}$, $p^R_{nc} = (r^*_{nc} - \epsilon)/\lambda_{nc}$. These are valid probabilities because of the assumption that $r^*_{nc} + \epsilon \leq \lambda_{nc}$ and $\epsilon \leq r^*_{nc}$ for all $(n,c) \in \mathbb{M}$. Hence, we have:

$$\mathbb{E}\left\{R^*_{nc}(t) \,\middle|\, U^{(c)}_n(t) \geq Q\right\} = r^*_{nc} - \epsilon \quad (31)$$

$$\mathbb{E}\left\{R^*_{nc}(t) \,\middle|\, U^{(c)}_n(t) < Q\right\} = r^*_{nc} + \epsilon \quad (32)$$

Using (31), (32), and (29) in the definition of $\delta_{nc}(t)$ (given in (23)), we have:

$$\mathbb{E}\left\{\delta^*_{nc}(t) \,\middle|\, U^{(c)}_n(t) \geq Q\right\} = \epsilon \quad (33)$$

$$\mathbb{E}\left\{-\delta^*_{nc}(t) \,\middle|\, U^{(c)}_n(t) < Q\right\} = \epsilon \quad (34)$$

Plugging (33), (34), (30), and (29) into the right hand side of the drift expression (28) yields:

$$\begin{aligned}\Delta(\underline{X}(t)) - V\mathbb{E}\left\{\sum_{nc} g_{nc}(\gamma_{nc}(t)) \mid \underline{X}(t)\right\} \leq N^2B \\ -\omega\sum_{nc}1^R_{nc}(t)e^{\omega(U_n^{(c)}(t)-Q)}\frac{\epsilon}{2} \\ -\omega\sum_{nc}1^L_{nc}(t)e^{\omega(Q-U_n^{(c)}(t))}\frac{\epsilon}{2} \\ -V\sum_{nc}g_{nc}(r^*_{nc}) \quad (35)\end{aligned}$$

Now note that:

$$\begin{aligned}N^2\frac{\omega\epsilon}{2} - \frac{\omega\epsilon}{2}\sum_{nc}1^L_{nc}(t)e^{\omega(U_n^{(c)}(t)-Q)} \\ -\frac{\omega\epsilon}{2}\sum_{nc}1^R_{nc}(t)e^{\omega(Q-U_n^{(c)}(t))} \geq 0\end{aligned}$$

which follows because $1_{nc}^L(t) = 1$ if and only if $1_{nc}^R(t) = 0$. Adding this non-negative quantity to the right hand side of (35) yields:

$$\Delta(\underline{X}(t)) - V\mathbb{E}\left\{\sum_{nc} g_{nc}(\gamma_{nc}(t))\,|\underline{X}(t)\right\} \leq N^2(B+\omega\epsilon/2) - \frac{\omega\epsilon}{2}\sum_{nc} e^{\omega(U_n^{(c)}(t)-Q)} - \frac{\omega\epsilon}{2}\sum_{nc} e^{\omega(Q-U_n^{(c)}(t))} - V\sum_{nc} g_{nc}(r_{nc}^*) \quad (36)$$

The above inequality is in the exact form for application of the Lyapunov drift lemma (Lemma 2). We thus have the following time average bounds:

$$\overline{\frac{1}{N^2}\sum_{nc} e^{\omega(U_n^{(c)}-Q)}} \leq \frac{N^2(B+\omega\epsilon/2)+Vg_{max}}{N^2\omega\epsilon/2} \quad (37)$$

$$\overline{\sum_{nc} e^{\omega(Q-U_n^{(c)})}} \leq \frac{N^2(B+\omega\epsilon/2)+Vg_{max}}{\omega\epsilon/2} \quad (38)$$

where the overbar notation denotes the $\limsup$ of the expected time average. Furthermore, (36) also implies:

$$\liminf_{t\to\infty} \frac{1}{t}\sum_{nc} g_{nc}(\overline{\gamma}_{nc}(t)) \geq \sum_{nc} g_{nc}(r_{nc}^*) - \frac{N^2(B+\omega\epsilon/2)}{V} \quad (39)$$

where $\overline{\gamma}_{nc}(t) \triangleq \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{\gamma_{nc}(\tau)\}$.

D. Deriving the Backlog Bound

Note by convexity of the function e^x together with Jensen's inequality, we have:

$$\overline{\frac{1}{N^2}\sum_{nc} e^{\omega(U_n^{(c)}-Q)}} \geq e^{\omega\left(\overline{\frac{1}{N^2}\sum_{nc}(U_n^{(c)}-Q)}\right)}$$

Taking the $\log$ of both sides of the above inequality and using (37) yields:

$$\overline{\frac{1}{N^2}\sum_{nc}(U_n^{(c)}-Q)} \leq \frac{\log\left(\frac{N^2(B+\omega\epsilon/2)+Vg_{max}}{N^2\omega\epsilon/2}\right)}{\omega} \quad (40)$$

Thus, for $Q = \frac{2}{\omega}\log(V)$, we have:

$$\overline{\frac{1}{N^2}\sum_{nc} U_n^{(c)}} \leq O(\log(V))$$

proving the congestion bound of Theorem 2 and demonstrating stability of all queues $U_n^{(c)}(t)$.

E. Deriving the Edge Probability Bound

Define $\alpha_{nc}^E(t) \triangleq Pr[U_n^{(c)}(t) < \mu_{max}^{out}]$, and define time average probabilities $\overline{\alpha}_{nc}^E(t)$ as follows:

$$\overline{\alpha}_{nc}^E(t) \triangleq \frac{1}{t}\sum_{\tau=0}^{t-1}\alpha_{nc}^E(\tau)$$

Note that:

$$\mathbb{E}\left\{e^{\omega(Q-U_n^{(c)}(t))}\right\} \geq \alpha_{nc}^E(t)e^{\omega(Q-\mu_{max}^{out})}$$

It follows that $\overline{e^{\omega(Q-U_n^{(c)})}} \geq \overline{\alpha_{nc}^E}e^{\omega(Q-\mu_{max}^{out})}$, where $\overline{\alpha_{nc}^E} \triangleq \limsup_{t\to\infty}\overline{\alpha}_{nc}^E(t)$. Using this inequality together with (38) yields:

$$\begin{aligned}\overline{\alpha_{nc}^E} &\leq \left[\frac{N^2(B+\omega\epsilon/2)+Vg_{max}}{\omega\epsilon/2}\right]e^{\omega\mu_{max}^{out}}e^{-\omega Q} \\ &= \left[\frac{N^2(B+\omega\epsilon/2)+Vg_{max}}{\omega\epsilon/2}\right]e^{\omega\mu_{max}^{out}}\frac{1}{V^2} \\ &= O(1/V) \end{aligned} \quad (41)$$

where we have used the fact that $Q = \frac{2}{\omega}\log(V)$.

F. Deriving the Utility Bound

Lemma 5: All virtual queues $Z_{nc}(t)$ are stable under UDOA.

Proof: A time average bound on the $Z_{nc}(t)$ queues can be derived in a manner similar to the bound for the $U_n^{(c)}(t)$ queues. Specifically, we can substitute the same control variables $R_{nc}^*(t)$, $\mu_{nb}^{*(c)}(t)$ into the drift expression (28), but further substitute $\hat{\gamma}_{nc}(t) = 0$ (instead of $\gamma_{nc}^*(t) = r_{nc}^*$). This yields a drift expression of the form $\Delta(\underline{X}(t)) \leq C - 2\sum_{nc} Z_{nc}(t)r_{nc}^*$, proving stability (details omitted for brevity). □

Recall that $\overline{r}_{nc}(t)$ and $\overline{\gamma}_{nc}(t)$ denote expected time averages of the $R_{nc}(\tau)$ and $\gamma_{nc}(\tau)$ variables during the first t slots.

Lemma 6: Under UDOA, we have:

$$\limsup_{t\to\infty}\max[\overline{\gamma}_{nc}(t) - \overline{r}_{nc}(t), 0] \leq \mu_{max}^{out}\overline{\alpha_{nc}^E}$$

Proof: Note that exactly $\mu_{nb}^{(c)}(t)$ bits of commodity c data are transmitted over link (n,b) during any timeslot t in which $U_n^{(c)}(t) \geq \mu_{max}^{out}$. Define the indicator function $1_{nc}^E(t)$ to take the value 1 if $U_n^{(c)}(t) < \mu_{max}^{out}$, and zero else. Assuming all queues are initially empty, the total commodity c bits transmitted by a given node must be less than or equal to the total bits that entered the node, and hence:

$$\sum_{\tau=0}^{t-1}\left[R_{nc}(\tau) + \sum_a \mu_{an}^{(c)}(\tau)\right] \geq \sum_{\tau=0}^{t-1}\left[\sum_b \mu_{nb}^{(c)}(\tau) - 1_{nc}^E(\tau)\mu_{max}^{out}\right]$$

Taking expectations and dividing by t yields:

$$\overline{r}_{nc}(t) + \sum_a \overline{\mu}_{an}^{(c)}(t) \geq \sum_b \overline{\mu}_{nb}(t) - \overline{\alpha}_{nc}^E(t)\mu_{max}^{out} \quad (42)$$

However, because all virtual queues are stable (Lemma 5), we have that the $\liminf$ expression (15) from Lemma 1 holds. Substituting (42) into (15) yields:

$$\liminf_{t\to\infty}\left[\overline{r}_{nc}(t) + \mu_{max}^{out}\overline{\alpha}_{nc}^E(t) - \overline{\gamma}_{nc}(t)\right] \geq 0 \quad (43)$$

The inequality (43) directly implies that:

$$\limsup_{t\to\infty}(\overline{\gamma}_{nc}(t) - \overline{r}_{nc}(t)) \leq \mu_{max}^{out}\overline{\alpha_{nc}^E}$$

which implies the conclusion of the lemma. □

Now define ν to be the largest right derivative of any utility function. Note that $\nu < \infty$ because we have assumed all utility

functions are concave with finite right derivatives. We thus have for all $(n,c) \in \mathbb{M}$:

$$\begin{aligned} g_{nc}(\overline{r}_{nc}(t)) &= g_{nc}\left(\overline{\gamma}_{nc}(t) - [\overline{\gamma}_{nc}(t) - \overline{r}_{nc}(t)]\right) \\ &\geq g_{nc}(\overline{\gamma}_{nc}(t)) - \nu \max[0, \overline{\gamma}_{nc}(t) - \overline{r}_{nc}(t)] \end{aligned}$$

We thus have:

$$\begin{aligned} \sum_{nc} g_{nc}(\overline{r}_{nc}(t)) &= \sum_{(n,c)\in\mathbb{M}} g_{nc}(\overline{r}_{nc}(t)) \\ &\geq \sum_{nc} g_{nc}(\overline{\gamma}_{nc}(t)) \\ &\quad -\nu \sum_{(n,c)\in\mathbb{M}} \max[0, \overline{\gamma}_{nc}(t) - \overline{r}_{nc}(t)] \end{aligned}$$

Taking the $\liminf$ and using Lemma 6 and inequality (41) yields:

$$\begin{aligned} \liminf_{t\to\infty} \sum_{nc} g_{nc}(\overline{r}_{nc}(t)) &\geq \liminf_{t\to\infty} \sum_{nc} g_{nc}(\overline{\gamma}_{nc}(t)) - O(1/V) \\ &\geq \sum_{nc} g_{nc}(r^*_{nc}) - O(1/V) \end{aligned}$$

where the final inequality follows by (39). This proves the utility bound and completes the proof of Theorem 2.

IX. Appendix B — Proof of Corollary 1

Proof: Let $x \triangleq \liminf_{t\to\infty} \sum_{nc} g_{nc}(\overline{r}_{nc}(t)) - g^*$. From Theorem 2, we have $x \geq -O(1/V)$ and hence:

$$x^3 \geq -O(1/V^3) \tag{44}$$

Now note that (37) in Appendix A together with Jensen's inequality and convexity of e^x imply that:

$$e^{\omega(\overline{U}-Q)} \leq O(V)$$

and so:

$$\begin{aligned} e^{-\omega\overline{U}} &\geq e^{-\omega Q} O(1/V) \\ &= \frac{1}{V^2} O(1/V) = O(1/V^3) \end{aligned} \tag{45}$$

where we have used $Q = (2/\omega)\log(V)$. Combining (45) and (44) yields $x^3 \geq -O(e^{-\omega\overline{U}})$, proving the result. □

References

[1] 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.

[2] M. J. Neely, E. Modiano, and C. Li. Fairness and optimal stochastic control for heterogeneous networks. *Proceedings of IEEE INFOCOM*, March 2005.

[3] M. J. Neely. Energy optimal control for time varying wireless networks. *Proceedings of IEEE INFOCOM*, March 2005.

[4] 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.

[5] M. J. Neely and E. Modiano. Improving delay in ad-hoc mobile networks via redundant packet transfers. *Proc. of the Conference on Information Sciences and Systems*, Johns Hopkins University: March 2003.

[6] M. J. Neely and E. Modiano. Capacity and delay tradeoffs for ad-hoc mobile networks. *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 1917-1937, June 2005.

[7] A. El Gammal, J. Mammen, B. Prabhakar, and D. Shah. Throughput-delay trade-off in wireless networks. *IEEE Proc. of INFOCOM*, 2004.

[8] S. Toumpis and A. J. Goldsmith. Large wireless networks under fading, mobility, and delay constraints. *IEEE Proceddings of INFOCOM*, 2004.

[9] X. Lin and N. B. Shroff. The fundamental capacity-delay tradeoff in large mobile ad hoc networks. *Purdue University Tech. Report*, 2004.

[10] F.P. Kelly, A.Maulloo, and D. Tan. Rate control for communication networks: Shadow prices, proportional fairness, and stability. *Journ. of the Operational Res. Society*, 49, p.237-252, 1998.

[11] S. H. Low. A duality model of tcp and queue management algorithms. *IEEE Trans. on Networking*, Vol. 11(4), August 2003.

[12] L. Xiao, M. Johansson, and S. Boyd. Simultaneous routing and resource allocation for wireless networks. *Proc. of the 39th Annual Allerton Conf. on Comm., Control, Comput.*, Oct. 2001.

[13] D. Julian, M. Chiang, D. O'Neill, and S. Boyd. Qos and fairness constrained convex optimization of resource allocation for wireless cellular and ad hoc networks. *Proc. INFOCOM*, 2002.

[14] P. Marbach. Priority service and max-min fairness. *IEEE Proceedings of INFOCOM*, 2002.

[15] P. Marbach and R. Berry. Downlink resource allocation and pricing for wireless networks. *IEEE Proc. of INFOCOM*, 2002.

[16] R. Berry, P. Liu, and M. Honig. Design and analysis of downlink utility-based schedulers. *Proceedings of the 40th Allerton Conference on Communication, Control, and Computing*, Oct. 2002.

[17] B. Krishnamachari and F. Ordonez. Analysis of energy-efficient, fair routing in wireless sensor networks through non-linear optimization. *IEEE Vehicular Technology Conference*, Oct. 2003.

[18] J. W. Lee, R. R. Mazumdar, and N. B. Shroff. Downlink power allocation for multi-class cdma wireless networks. *IEEE Proceedings of INFOCOM*, 2002.

[19] R. Cruz and A. Santhanam. Optimal routing, link scheduling, and power control in multi-hop wireless networks. *IEEE Proceedings of INFOCOM*, April 2003.

[20] M. Chiang. To layer or not to layer: Balancing transport and physical layers in wireless multihop networks. *IEEE Proceedings of INFOCOM*, March 2004.

[21] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transacations on Automatic Control*, Vol. 37, no. 12, Dec. 1992.

[22] L. Tassiulas and A. Ephremides. Dynamic server allocation to parallel queues with randomly varying connectivity. *IEEE Trans. on Inform. Theory*, vol. 39, pp. 466-478, March 1993.

[23] P.R. Kumar and S.P. Meyn. Stability of queueing networks and scheduling policies. *IEEE Trans. on Automatic Control*, Feb. 1995.

[24] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, and P. Whiting. Providing quality of service over a shared wireless link. *IEEE Communications Magazine*, 2001.

[25] M. J. Neely, E. Modiano, and C. E. Rohrs. Power allocation and routing in multi-beam satellites with time varying channels. *IEEE Transactions on Networking*, Feb. 2003.

[26] 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*, January 2005.

[27] E. M. Yeh and A. S. Cohen. Throughput optimal power and rate control for multiaccess and broadcast communications. *Proc. of the International Symposium on Information Theory*, June 2004.

[28] X. Liu, E. K. P. Chong, and N. B. Shroff. A framework for opportunistic scheduling in wireless networks. *Computer Networks*, vol. 41, no. 4, pp. 451-474, March 2003.

[29] A. Eryilmaz and R. Srikant. Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control. *IEEE Proceedings of INFOCOM*, March 2005.

[30] A. Stolyar. Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm. *Queueing Systems*, 2005.

[31] J. W. Lee, R. R. Mazumdar, and N. B. Shroff. Opportunistic power scheduling for dynamic multiserver wireless systems. *to appear in IEEE Trans. on Wireless Systems*, 2005.

[32] M. J. Neely. Optimal energy and delay tradeoffs for multi-user wireless downlinks. *Proceedings of IEEE INFOCOM*, April 2006.

[33] F. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 1997.

[34] A. Tang, J. Wang, and S. Low. Is fair allocation always inefficient. *IEEE Proceedings of INFOCOM*, March 2004.

[35] M. J. Neely. Optimal energy and delay tradeoffs for multi-user wireless downlinks. *USC Technical Report CSI-05-06-01*, June 2005.