Fairness and Optimal Stochastic Control for Heterogeneous Networks

Michael J. Neely, *Member, IEEE*, Eytan Modiano, *Senior Member, IEEE*, and Chih-Ping Li

***Abstract*—We consider optimal control for general networks with both wireless and wireline components and time varying channels. A dynamic strategy is developed to support all traffic whenever possible, and to make optimally fair decisions about which data to serve when inputs exceed network capacity. The strategy is decoupled into separate algorithms for flow control, routing, and resource allocation, and allows each user to make decisions independent of the actions of others. The combined strategy is shown to yield data rates that are arbitrarily close to the optimal operating point achieved when all network controllers are coordinated and have perfect knowledge of future events. The cost of approaching this fair operating point is an end-to-end delay increase for data that is served by the network.**

***Index Terms*—Distributed computing, flow control, queueing analysis, satellite networks, stochastic optimization, wireless networks.**

I. Introduction

Modern data networks consist of a variety of heterogeneous components, and continue to grow as new applications are developed and new technologies are integrated into the existing communication infrastructure. While network resources are expanding, the demand for these resources is also expanding, and it is often the case that data links are loaded with more traffic than they were designed to handle. In order to provide high speed connectivity for future personal computers, hardware devices, wireless units, and sensor systems, it is essential to develop fair networking techniques that take full advantage of all resources and system capabilities. Such techniques must be implemented through simple, localized message passing protocols between neighboring network elements.

In this paper, we design a set of decoupled algorithms for resource allocation, routing, and flow control for general networks with both wireless and wireline data links and time varying channels. Specifically, we treat a network with N nodes and L links. The condition of each link at a given time t is described by a *link state vector* $\vec{S}(t) = (S_1(t), \ldots, S_L(t))$, where $S_l(t)$ is a parameter characterizing the communication channel for link l. For example, if l is a wireless link, $S_l(t)$ may represent the current attenuation factor or noise level. In an unreliable wired link, $S_l(t)$ may take values in the two-element set $\{ON, OFF\}$, indicating whether link l is available for communication. We consider a slotted system model with slots normalized to integral units $t \in \{0, 1, 2, \ldots\}$. Channels hold their state for the duration of a timeslot, and potentially change states on slot boundaries. For simplicity of exposition, we assume there are a finite (but arbitrarily large) number of possible channel state vectors, and that the vectors $\vec{S}(t)$ are independent and identically distributed (i.i.d.) over timeslots. We note that the algorithms we design under this i.i.d. assumption can be applied to yield similar performance for systems with general ergodic channel state variations.

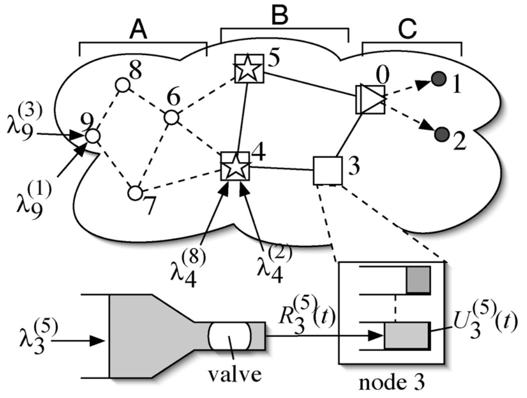

Fig. 1. (a) A heterogeneous network with wireless and wireline data links, and (b) a close-up of one node, illustrating the internal queues and the storage reservoir for exogenous arrivals.

For each channel state $\vec{S}$, let $\Gamma_{\vec{S}}$ denote the set of link transmission rates available for resource allocation decisions when $\vec{S}(t) = \vec{S}$. In particular, every timeslot t the network controllers are constrained to choosing a transmission rate vector $\vec{\mu}(t) = (\mu_1(t), \ldots, \mu_L(t))$ such that $\vec{\mu}(t) \in \Gamma_{\vec{S}(t)}$ (where $\mu_l(t)$ is the transmit rate over link l and has units of bits/slot). Use of this abstract set of transmission rates $\Gamma_{\vec{S}}$ maintains a simple separation between network layer and physical layer concepts, yet is general enough to allow network control to be suited to the unique capabilities of each data link. We assume all sets $\Gamma_{\vec{S}}$ are compact (closed and bounded).

As an example, consider the heterogeneous network of Fig. 1 consisting of three separate groups of links A, B, and C: Set A represents a wireless sensor system that connects to a wired infrastructure through two uplink access points, set B represents the wired data links, and set C represents the two downlink channels of a basestation that transmits to two different users

Manuscript received March 28, 2005; revised February 7, 2006, and July 23, 2006; approved by IEEE/ACM Transactions on Networking Editor I. Stavrakakis. This work was supported in part by one or more of the following: NSF ITR grant CCR-0325401, ONR grant number N000140610064, DARPA/AFOSR through the University of Illinois grant no. F49620-02-1-0325, DARPA IT-MANET grant W911NF-07-0028, NSF grant OCE 0520324. This work was presented in part at the IEEE INFOCOM Conference, Miami, FL, March 2005, and is based on the Ph.D. dissertation of the first author (Massachusetts Institute of Technology, 2003).

M. J. Neely and C. Li are with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089 USA (e-mail: mjneely@usc.edu; web: http://www-rcf.usc.edu/~mjneely).

E. Modiano is with the Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: modiano@mit.edu).

Digital Object Identifier 10.1109/TNET.2007.900405

1 and 2. For a given channel state $\vec{S}$, the set of feasible transmission rates $\Gamma_{\vec{S}}$ reduces to a product of rates corresponding to the three independent groups:

$$\Gamma_{\vec{S}} = \Gamma^{A}_{\vec{S}_A} \times \Gamma^{B} \times \Gamma^{C}_{\vec{S}_C}. \tag{1}$$

Set $\Gamma^{A}_{\vec{S}_A}$ might contain a continuum of link rates associated with the channel interference properties and power allocation options of the sensor nodes, and depends only on the link states $\vec{S}_A$ of these nodes. Set Γ^{B} might contain a single vector $(C_1, \ldots, C_k)$ representing the fixed capacities of the k wired links. Set $\Gamma^{C}_{\vec{S}_C}$ might represent a set of two vectors $\{(\phi_{S_1}, 0), (0, \phi_{S_2})\}$, where ϕ_{S_i} is the rate available over link i if this link is selected to transmit on the given timeslot t when $S_i(t) = S_i$. We further note that the ground network shown in the figure can be augmented to include a collection of *satellite nodes*, increasing route diversity and enabling connectivity to other sub-networks.

Data is transmitted from node to node over potentially multi-hop paths to reach its destination. Data arrives to the network according to random processes with well-defined time average rates, and we let $\lambda_n^{(c)}$ represent the long term average rate of new exogenous arrivals to source node n intended for destination node c (in units of bits/slot). Let $(\lambda_n^{(c)})$ be the matrix of exogenous arrival rates. The *network layer capacity region* Λ is defined as the closure of the set of all arrival matrices that are stably supportable by the network, considering all possible multi-hop routing and resource allocation policies (possibly those with perfect knowledge of future events). We emphasize that, while the channel states may change from slot to slot, the set Λ is fixed and depends only on the steady state channel probabilities and the link transmission rate sets $\Gamma_{\vec{S}}$ [1], [28].

The set Λ can be shown to be compact and convex [1]. In [28], a routing and power allocation policy was developed to stabilize a general wireless network whenever the rate matrix $(\lambda_n^{(c)})$ is within the capacity region Λ. The purpose of our current paper is to treat heterogeneous networks and develop distributed algorithms for flow control, routing, and resource allocation that provide optimal fairness in cases when arrival rates are *either inside or outside the network capacity region*.

Specifically, we define a set of *utility functions* $g_n^{(c)}(r)$, representing the "satisfaction" received by sending data from node n to node c at a time average rate of r bits/slot. The utility functions are assumed to be non-decreasing and concave. The goal is to support a fraction of the traffic demand matrix $(\lambda_n^{(c)})$ to achieve long term throughputs $(\overline{r}_n^{(c)})$ that maximize the sum of user utilities. The *optimal sum utility* is thus defined by the following optimization problem:

$$\text{Maximize :} \quad \sum_{n,c} g_n^{(c)}\left(\overline{r}_n^{(c)}\right) \tag{2}$$

$$\text{Subject to :} \quad \left(\overline{r}_n^{(c)}\right) \in \Lambda \tag{3}$$

$$0 \leq \overline{r}_n^{(c)} \leq \lambda_n^{(c)} \text{ for all } (n,c). \tag{4}$$

Inequality (3) is the *stability constraint* and ensures that the long term admitted rates are stabilizable by the network. Inequality (4) is the *demand constraint* that ensures the rate provided to session (n,c) is no more than the incoming traffic rate of this session.

Because the functions $g_n^{(c)}(r)$ are non-decreasing, it is clear that if $(\lambda_n^{(c)}) \in \Lambda$, then the above optimization is solved by the matrix $(r_n^{*(c)}) = (\lambda_n^{(c)})$. If $(\lambda_n^{(c)}) \notin \Lambda$, then the solution $(r_n^{*(c)})$ will lie somewhere on the capacity region boundary. The above optimization could in principle be solved if the arrival rates $(\lambda_n^{(c)})$ and the capacity region Λ were known in advance, and all users could coordinate by sending data according to the optimal solution. However, the capacity region depends on the channel probabilities, which are unknown to the network controllers and to the individual users. Furthermore, the individual users do not know the data rates or utility functions of other users. In this paper, we develop a practical dynamic control strategy that yields a resulting set of throughputs $(\overline{r}_n^{(c)})$ that are arbitrarily close to the optimal solution of (2)–(4). The distance to the optimal solution is shown to decrease like $1/V$, where V is a control parameter affecting a tradeoff in average delay for data that is served by the network.

Previous work on network fairness and optimization is found in [5]–[17]. Utility optimization problems similar to (2), (3) are considered for static wireless downlinks with infinite backlog in [5], and for static multi-hop wireless networks in [8]. Further static resource allocation problems for wireless and wireline systems are treated in [6]–[17]. Much of this work uses convex optimization and Lagrangian duality to achieve a fixed resource allocation that is optimal with respect to various utility metrics. In [12], [13], distributed pricing mechanisms are used to provide *proportional fairness* in static flow networks. Control laws based on continuous time differential equations are used in [13], [16] to ensure flows converge to a utility optimal operating point, and dual sub-gradient methods are considered in [14]. The relationship between duality theory, utility optimization, and classical internet congestion control techniques is explored in [17].

We note that fixed allocation solutions may not be appropriate in cases when optimal control involves *dynamic resource allocation*. Dynamic policies might be required for optimality even when the underlying network is static [19]. The capacity of a multi-user wireless downlink with *randomly varying channels* is established in [34], and utility optimization in a similar system is treated in [18]. These formulations do not consider stochastic arrivals and queueing, and the resulting algorithms are developed and analyzed under the assumption that channel probabilities are fully known.

The design of dynamic controllers to stabilize stochastic wireless and wireline networks is considered in [20]–[28] using a powerful theory of *Lyapunov drift*. However, this work primarily addresses queueing stability and does not provide methods for achieving both stability and performance optimization, as required to solve the general fairness problem (2)–(4) for stochastic networks. Dynamic algorithms for fair scheduling in wireless downlinks are addressed in [30]–[32], but do not yield optimal performance for all input rates, as discussed in the next section. A special case solution of (2)–(4) is developed in [33] for a class of wireless downlinks with linear utility functions and ON/OFF channels.

The main contribution of our work is the development of a novel control policy that yields optimal performance for general stochastic networks and general fairness metrics. The policy does not require knowledge of channel statistics, input rates, or the global network topology. Our analysis establishes a new and important Lyapunov drift technique that enables stability and performance optimization to be achieved simultaneously, extending the stability results developed in [20]–[28]. This work presents a fundamental approach to *stochastic network optimization* [1]–[4]. We note that alternative optimization approaches have recently been considered in [29], [35] using fluid limit models, and in [36] using stochastic gradient theory (see, for example, [37]). Our Lyapunov optimization technique is related to the theory of static and stochastic gradients (see [1, Ch. 4.7–5.7]). However, our techniques were developed within the framework of Lyapunov stability theory for queueing systems, and yield explicit network utility and delay guarantees.

In the next section, we illustrate the challenges of stochastic network optimization with a simple downlink example. In Section III we develop a fair scheduling algorithm for general networks under the special case when all active input reservoirs are "infinitely backlogged." In Section V we construct a modified algorithm that yields optimal performance without the infinite backlog assumption. This is another important contribution of our paper, and our solution illuminates the differences between optimizing an expectation and optimizing a concave function of an expectation. Example simulations for wireless networks and $N \times N$ packet switches are presented in Section VI, and distributed implementation strategies for wireless networks are discussed in Section VII.

II. Satellite Downlink Example

Consider a satellite node (or wireless basestation) that transmits data to two downlink users 1 and 2 over two different channels (as illustrated by considering only the triangle-node of the network in Fig. 1). Time is slotted and packets for each user arrive to the basestation according to independent Bernoulli processes with rates λ_1 and λ_2. Let $U_1(t)$ and $U_2(t)$ represent the current backlog of packets waiting for transmission to user 1 and user 2, respectively. Channels independently vary between ON and OFF states every slot according to Bernoulli processes, with ON probabilities p_1 and p_2, and we assume that $p_1 < p_2$. Every timeslot, a controller observes the channel states and chooses to transmit over either channel 1 or channel 2. We assume that a single packet can be transmitted if a channel is ON and no packet can be transmitted when a channel is OFF, so that the only decision is which channel to serve when both channels are ON.

The capacity region Λ for this system is described by the set of all rates (λ_1, λ_2) that satisfy

$$\lambda_1 \leq p_1, \quad \lambda_2 \leq p_2, \quad \lambda_1 + \lambda_2 \leq p_1 + (1 - p_1)p_2.$$

These conditions are necessary for stability because the long term output rate from any channel i is at most p_i, and the maximum sum rate out of the system is $p_1 + (1 - p_1)p_2$. Furthermore, it is shown in [21] that the 'Maximum Weight Match' (MWM) policy of serving the ON queue with the largest backlog achieves stability whenever input rates are strictly interior to the above region.

Now define $g_1(r) = g_2(r) = \log(r)$, and consider the *proportional fairness* control objective of maximizing $\log(r_1) + \log(r_2)$, where r_1 and r_2 are the delivered throughputs over channels 1 and 2 (see [12] for a discussion of proportional fairness). We evaluate three well-known algorithms with respect to this fairness metric: The Borst algorithm [30], the 'proportionally fair' Max μ_i/r_i algorithm [31], [32], and the MWM policy [21].

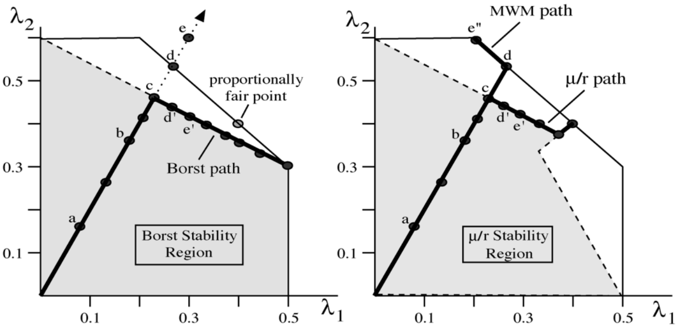

Fig. 2. The downlink capacity region Λ and the stability regions of the Borst policy and the Max μ_i/r_i policy. Input rates (λ_1, λ_2) are pushed toward point (0.5, 1.0), and the simulated throughputs under the Borst, Max μ_i/r_i, and MWM policies are illustrated.

The Borst algorithm chooses the nonempty channel i with the largest $\mu_i(t)/\overline{\mu}_i$ index, where $\mu_i(t)$ is the current transmission rate offered over channel i, and $\overline{\mu}_i$ is the average of $\mu_i(t)$. This algorithm is shown in [30] to provide optimal fairness for wireless networks with an "infinite" number of channels, where each incoming packet is destined for a unique user with its own channel. Although the algorithm was not designed for the 2-queue downlink described above, it is closely related to the Max μ_i/r_i policy, and it is illuminating to evaluate its performance in this context. Applied to the 2-queue downlink, the Borst algorithm reduces to serving the nonempty ON queue with the largest value of $1/p_i$. Because $p_1 < p_2$, this algorithm effectively gives packets destined for channel 1 strict priority over channel 2 packets. Thus, the service of queue 1 is independent of the state of channel 2, and conditioning on the event that a packet is served from channel 1 during a particular timeslot does not change the probability that channel 2 is ON. It follows that the rate of serving channel 1 packets while channel 2 is ON is given by $\lambda_1 p_2$ (assuming queue 1 is stable so that all λ_1 traffic is served). Thus, the stability region of the Borst algorithm is given by

$$\lambda_1 \leq p_1, \quad \lambda_2 \leq p_2 - \lambda_1 p_2 \tag{5}$$

which is a strict subset of the capacity region (see Fig. 2).

Consider now the related policy of serving the nonempty queue with the largest value of $\mu_i(t)/r_i(t)$, where $r_i(t)$ is the empirical throughput achieved over channel i. This differs from the Borst algorithm in that transmission rates are weighted by the throughput actually delivered rather than the average transmission rate that is offered. This Max μ_i/r_i policy is proposed in [31], [32] and shown to have desirable proportional fairness properties when all queues of the downlink are infinitely backlogged. To evaluate its performance for arbitrary traffic rates (λ_1, λ_2), suppose the running averages $r_1(t)$ and $r_2(t)$ are accumulated over the entire timeline, and suppose the system is stable so that $r_1(t)$ and $r_2(t)$ converge to λ_1 and λ_2. It follows that the algorithm eventually reduces to giving channel 1 packets strict priority if $\lambda_1 < \lambda_2$, and giving channel 2 packets strict priority if $\lambda_2 < \lambda_1$. Thus, if $\lambda_1 < \lambda_2$ then these rates must also satisfy the inequalities (5), while $\lambda_2 < \lambda_1$ implies the rates must satisfy the inverted inequalities $\lambda_2 \leq p_2$ and $\lambda_1 \leq p_1 - \lambda_2 p_1$. Thus, at first glance it seems that the stability region of this policy is a subset of the stability region of the Borst algorithm. However, its stability region has the peculiar property of including all feasible rate pairs (λ, λ) (see Fig. 2).

In Fig. 2, we consider the special case when $p_1 = 0.5$, $p_2 = 0.6$, and plot the achieved throughput of the Borst, Max μ_i/r_i, and MWM policies when the rate vector (λ_1, λ_2) is scaled linearly towards the vector (0.5, 1.0), illustrated by the ray in Fig. 2(a). One hundred different rate points on this ray were considered (including example points $a - e$), and simulations were performed for each point over a period of 5 million timeslots. Fig. 2(a) illustrates the resulting throughput of the Borst algorithm, where we have included example points d' and e' corresponding to input rate points d and e. Note that the Borst algorithm always results in throughput that is strictly interior to the capacity region, even when input rates are outside of capacity. Fig. 2(b) illustrates performance of the Max μ_i/r_i and MWM policies. Note that the MWM policy supports all (λ_1, λ_2) traffic when this rate vector is within the capacity region. However, when traffic is outside of the capacity region the achieved throughput moves along the boundary in the wrong direction, yielding throughputs that are increasingly unfair because it favors service of the higher traffic rate stream. Like the Borst policy, the Max μ_i/r_i policy leads to instability for all (stabilizable) input rates on the ray segment $c - d$, and yields throughput that is strictly interior to the capacity region even when inputs exceed system capacity (compare points e and e'). However, the throughput eventually touches the capacity region boundary, reaching the proportionally fair point (0.4, 0.4) when input rates are sufficiently far outside of the capacity region.

It is clear from this simple downlink example that there is a need for a *universally fair* algorithm, one that performs well regardless of whether inputs are inside or outside of the capacity region. For this example, such an algorithm would yield throughput that increases toward the point d of the figure, and then moves on the boundary of the capacity region toward the fair operating point thereafter. In the following, we develop such an algorithm for general multihop networks.

III. Control of Heterogeneous Networks

Consider a heterogeneous network with N nodes, L links, and time varying channels $\vec{S}(t)$ (see example in Fig. 1). Each link $l \in \{1, \ldots, L\}$ represents a directed communication channel for transmission from one node to another, and we define $tran(l)$ and $rec(l)$ as the corresponding transmitting and receiving nodes, respectively. Each node of the network maintains a set of output queues for storing data according to its destination. All data (from any source node) that is destined for a particular node $c \in \{1, \ldots, N\}$ is classified as *commodity c data*, and we let $U_n^{(c)}(t)$ represent the backlog of commodity c data currently stored in the network layer at node n (see Fig. 1). A *network layer control algorithm* makes decisions about routing, scheduling, and resource allocation in reaction to current channel state and queue backlog information. The objective is to deliver all data to its proper destination, potentially by routing over multi-hop paths.

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 $\mathcal{L}_c$ as the set of all data links l that are acceptable for commodity c data to traverse. As a general algorithm might schedule multiple commodities to flow over the same link on a given timeslot, we define $\mu_l^{(c)}(t)$ as the rate offered to commodity c traffic along link l during timeslot t.[1] The transmission rates and routing variables are chosen by a *dynamic scheduling and routing algorithm*. Specifically, the network makes the following control decisions every slot:

- *Resource (Rate) Allocation*: Choose a transmission rate vector $\vec{\mu}(t) = (\mu_1(t), \ldots, \mu_L(t))$ such that $\vec{\mu}(t) \in \Gamma_{\vec{S}(t)}$
- *Routing/Scheduling*: For each link l and each commodity c, choose $\mu_l^{(c)}(t)$ to satisfy the following constraints:

$$\sum_c \mu_l^{(c)}(t) \leq \mu_l(t) \tag{6}$$

$$\mu_l^{(c)}(t) = 0 \quad \text{if } l \notin \mathcal{L}_c. \tag{7}$$

We note that defining all sets $\mathcal{L}_c$ to be equal to the set of all network links effectively removes the routing constraints (7) and reduces to the case of *unconstrained routing*. This creates the most options, although it is often useful to constrain routes via the $\mathcal{L}_c$ sets to ensure more predictable performance and maintain FIFO or near FIFO delivery. For example, in cases where the network topology is fixed and it is desirable to route all data associated with a particular source-destination pair over a single path, then the sets $\mathcal{L}_c$ can be defined as a *tree* of links from each source to the destination c. In cases where it is preferred to have two or more different sources of the same commodity use paths that cross but do not merge, then this single "commodity" can be redefined as several commodities to distinguish the different source-destination pairs (which also increases the total number of distinct queues $U_n^{(c)}(t)$ in the network).

A set of *flow controllers* act at every node to limit the new data admitted into the network layer. Specifically, new data of commodity c that arrives to source node n is first placed in a *transport layer storage reservoir* (n, c). A control valve determines the amount of data $R_n^{(c)}(t)$ released from this reservoir on each timeslot (see Fig. 1). The $R_n^{(c)}(t)$ process acts as the exogenous arrival process to the queue $U_n^{(c)}(t)$. Endogenous arrivals consist of commodity c data transmitted to node n from other network nodes. Define Ω_n as the set of all links l such that $tran(l) = n$, and define Θ_n as the set of all links such that $rec(l) = n$. Every timeslot the backlog $U_n^{(c)}(t)$ changes according to the following queueing law:

$$U_n^{(c)}(t+1) \leq \max\left[U_n^{(c)}(t) - \sum_{l \in \Omega_n} \mu_l^{(c)}(t), 0\right] + \sum_{l \in \Theta_n} \mu_l^{(c)}(t) + R_n^{(c)}(t). \tag{8}$$

The expression above is an inequality rather than an equality because the endogenous arrivals may be less than $\sum_{l \in \Theta_n} \mu_l^{(c)}(t)$ if nodes have little or no commodity c data to transmit. The above dynamics hold for all node pairs $n \neq c$. Data leaves the network

[1] We find that the capacity achieving solution needs only route a single commodity over any given link during a timeslot.

when it reaches its destination, and so we define $U_n^{(n)}(t) \triangleq 0$ for all n and all t.

Let $A_n^{(c)}(t)$ represent the new commodity c data arriving to the system at source node n during slot t, and let $L_n^{(c)}(t)$ represent the current backlog in the flow control reservoir at time t. The control decision variables $R_n^{(c)}(t)$ are chosen every timeslot according to the following restrictions:

- *Flow Control*: Choose $R_n^{(c)}(t)$ such that

$$R_n^{(c)}(t) \leq L_n^{(c)}(t) + A_n^{(c)}(t) \text{ for all } t$$
$$\sum_c R_n^{(c)}(t) \leq R_n^{max} \text{ for all } t.$$

where the constants R_n^{max} are chosen to be positive and suitably large, to be made precise in the development of our two different flow control strategies CLC1 and CLC2. The first flow control constraint ensures that admitted data is less than or equal to the actual data available, and the second is important for limiting the burstiness of the admitted arrivals.

A. *The Network Capacity Region*

Due to the routing constraints (7), some commodities might never be able to visit certain nodes. Further, some nodes might only be associated with destinations, and hence these nodes do not keep any internal queues. Hence, we define K_n as the number of internal queues kept by node n, where $K_n \in \{0, 1, \ldots, N-1\}$. Define $\mathcal{D}$ as the set of all node-commodity pairs (n, c) associated with internal queues in the network, and let D represent the number of such queues:

$$D \triangleq \sum_{n=1}^{N} K_n.$$

The integer D defines the *relative dimension* of the network. We assume that all active traffic sessions are within the set $\mathcal{D}$, so that $R_n^{(c)}(t) \triangleq 0$, $g_n^{(c)}(r) \triangleq 0$ for all $(n, c) \notin \mathcal{D}$. We further define $U_n^{(c)}(t) \triangleq 0$ for all t when $(n, c) \notin \mathcal{D}$.

Suppose the admitted traffic from each flow control valve (n, c) has a well-defined time average $\overline{r}_n^{(c)} \triangleq \lim_{t\to\infty}(1/t)\sum_{\tau=0}^{t-1}\mathbb{E}\{R_n^{(c)}(\tau)\}$. The *network layer capacity region* Λ is the set of all time average rate matrices $(\overline{r}_n^{(c)})$ that the network can stably support, considering all possible routing and resource allocation algorithms that satisfy the constraints (6), (7) and the queueing dynamics (8) described above. Stability of a queue with general arrival and transmission rate processes is defined as follows:

Definition 1: A queue $U(t)$ with general stochastic arrival and transmission rate processes is *strongly stable* if

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{U(\tau)\right\} < \infty.$$

That is, we define strong stability (hereafter called "stability") in terms of a finite time average backlog.

In [1], [28], the network capacity region Λ is characterized for the special case of unconstrained routing, where the constraints (7) are removed. It is not difficult to extend this result to include routing constraints (7). We thus state the following modified version of the network capacity theorem from [1], [28], omitting the proof for brevity.

Theorem 1 (Network Capacity): If channels $\vec{S}(t)$ are i.i.d. over slots, a fixed input rate matrix $(r_n^{(c)})$ is in the capacity region Λ if and only if $r_n^{(c)} = 0$ for all $(n, c) \notin \mathcal{D}$ and there exists a stationary randomized resource allocation algorithm that chooses particular transmission rates $\mu_l^{*(c)}(t)$ based only on the current channel state $\vec{S}(t)$, and satisfies for all slots t

$$\mathbb{E}\left\{\sum_{l\in\Omega_n}\mu_l^{*(c)}(t) - \sum_{l\in\Theta_n}\mu_l^{*(c)}(t)|\underline{U}(t)\right\} = r_n^{(c)}, \quad \forall n \neq c$$
$$\mu_l^{*(c)}(t) = 0, \quad \text{if } tran(l) = c \text{ or if } l \notin \mathcal{L}_c$$

where $\underline{U}(t) = (U_n^{(c)}(t))$ represents the matrix of current queue backlogs.

The above expectation is taken over the probability distribution of the current channel state $\vec{S}(t)$ and the distribution of the randomized resource allocation decisions that depend on this channel state. Because channels are i.i.d., the above expectation is the same every slot t, and does not depend on $\underline{U}(t)$. The expectation is intentionally conditioned on $\underline{U}(t)$ as the existence of a policy that satisfies this exact equality will be used later. It can be shown that the capacity region Λ depends only on the steady state channel probability distribution [1], [28], and hence the region Λ described by Theorem 1 is unchanged if actual channels are non-i.i.d. but are ergodic with the same steady sate distribution as in the i.i.d. case.

The capacity region Λ can be shown to be compact and convex with D effective dimensions (see [1] for a similar result). It shall be useful to define the parameter μ_{sym} to be the largest rate that is simultaneously supportable by all sessions $(n, c) \in \mathcal{D}$, so that $(\mu_{sym}1_n^{(c)}) \in \Lambda$ (where $1_n^{(c)}$ is equal to 1 if $(n, c) \in \mathcal{D}$, and zero else). Geometrically, the value μ_{sym} represents the edge size of the largest D-dimensional hypercube that can be fit into the capacity region Λ, and is a value that unexpectedly arises in our analysis. We assume throughout that $\mu_{sym} > 0$.

B. *Dynamic Control for Infinite Demand*

Here we develop a practical control algorithm that stabilizes the network and ensures that utility is arbitrarily close to optimal, with a corresponding tradeoff in network delay. Recall that functions $g_n^{(c)}(r)$ represent the utility of supporting rate r communication from node n to node c (we define $g_n^{(c)}(r) = 0$ if there is no active session of traffic originating at node n and destined for node c). To highlight the fundamental issues of routing, resource allocation, and flow control, in this section we assume that all active sessions (n, c) have *infinite backlog* in their corresponding reservoirs, so that flow variables $R_n^{(c)}(t)$ can be chosen without first establishing that this much data is available in the reservoir. Flow control is imperative in this infinite backlog scenario, and the resulting problem is simpler as it does not involve the demand constraint (4). A modified algorithm is developed in Section V for the general case of finite demand matrices $(\lambda_n^{(c)})$ and finite buffer reservoirs.

The following control strategy is decoupled into separate algorithms for resource allocation, routing, and flow control. The strategy combines a novel flow control technique together with a generalization of the Dynamic Routing and Power Control (DRPC) strategy of [28].

Cross-Layer Control Algorithm 1 (CLC1)

- *Flow Control*—(algorithm FLOW) Every timeslot t, the flow controller at each node n observes the current level of queue backlogs $U_n^{(c)}(t)$ for each commodity $c \in \{1, \ldots, N\}$. It then chooses $R_n^{(c)}(t)$ as the solution of an optimization problem. Specifically, it chooses $R_n^{(c)}(t) = x_n^{(c)}$, where the $x_n^{(c)}$ values solve the following:

$$\text{Maximize :} \quad \sum_{c=1}^{N} \left[V g_n^{(c)}\left(x_n^{(c)}\right) - 2x_n^{(c)} U_n^{(c)}(t) \right] \tag{9}$$

$$\text{Subject to :} \quad \left(x_n^{(c)}\right) \geq 0, \quad \sum_{c=1}^{N} x_n^{(c)} \leq R_n^{max} \tag{10}$$

  where $V > 0$ is a chosen constant that affects the performance of the algorithm. Note that $R_n^{(c)}(t)$ can change every slot t according to the optimization (9)–(10). The $x_n^{(c)}$ values are dummy variables used only to define the optimization, and may be different for different slots t.
- *Routing and Scheduling*—Each node n observes the backlog in all neighboring nodes j to which it is connected by a link l (where $tran(l) = n$, $rec(l) = j$). Let $W_l^{(c)}(t) = U_{tran(l)}^{(c)}(t) - U_{rec(l)}^{(c)}(t)$ represent the *differential backlog* of commodity c data. Define $W_l^*(t) \triangleq \max_{[c|l \in \mathcal{L}_c]}\{W_l^{(c)}(t), 0\}$ as the maximum differential backlog over link l (maxed with 0), and let $c_l^*(t)$ represent the maximizing commodity. Data of commodity $c_l^*(t)$ is selected for (potential) routing over link l whenever $W_l^*(t) > 0$.
- *Resource Allocation*—The current channel state $\vec{S}(t)$ is observed, and a transmission rate vector $\vec{\mu}(t)$ is selected by maximizing $\sum_l W_l^*(t)\mu_l(t)$ subject to the constraint $\vec{\mu}(t) \in \Gamma_{\vec{S}(t)}$. The resulting transmission rate of $\mu_l(t)$ is offered to commodity $c_l^*(t)$ data on link l (provided that $W_l^*(t) > 0$). If any node does not have enough bits of a particular commodity to send over all outgoing links requesting that commodity, *null bits* are delivered.

The flow control algorithm is decentralized, where the control valves for each node n require knowledge only of the queue backlogs in node n. We note that the constraint $\sum_c R_n^{(c)}(t) \leq R_n^{max}$ (for all n) in the flow control optimization (9) can be replaced with the simpler and less restrictive constraint $R_n^{(c)}(t) \leq R_n^{max}$ (for all (n, c)), allowing for each $R_n^{(c)}(t)$ value to be obtained by maximizing a concave function of one variable. However, this would increase the delay bound (presented in Theorem 2 of the next subsection) roughly by a factor equal to the largest number of distinct commodities that are sourced at any single node. An appropriate value to assign the R_n^{max} constant is discussed in Section III-D.

The routing and scheduling algorithm acts according to a differential backlog strategy similar to the backpressure strategy developed in [20], and is decentralized provided that each node i knows the backlog levels of its neighbors. The resource allocation strategy of maximizing $\sum_l W_l^*(t)\mu_l(t)$ is the most complex part of the algorithm, but can be distributed over the independent portions of the network (as in (1)), or can be approximated as described in Section VII.

C. Intuitive Description of the Policy

The flow control policy (9) uses a parameter V that determines the extent to which utility optimization is emphasized. Indeed, if V is large relative to the current backlog in the source queues, then the admitted rates $R_n^{(c)}(t)$ will be large, increasing the time average utility while consequently increasing congestion. This effect is mitigated as backlog grows at the source queues and flow control decisions become more conservative. The routing and scheduling algorithm uses backpressure from neighboring nodes to route in directions of the largest *differential backlog*. Below we show that the algorithm can be used to drive network utility arbitrarily close to optimal by suitably increasing the V parameter, with a corresponding increase in network congestion. Intuitively, congestion grows because more "learning time" is required to achieve a finer and finer optimization. The queue backlogs and backlog differentials will help the controllers learn "good" directions to route and "good" amounts of data to admit, and the "noise" of fluctuating queues will have less influence when queue sizes are large. We note that in our more recent work [38], we characterize the fundamental tradeoff between utility and delay.

D. Algorithm Performance

To analyze the performance of the above CLC1 algorithm, we define the maximum transmission rates out of and into a given node n as follows:

$$\mu_{max,n}^{out} \triangleq \max_{[\vec{S}, \vec{\mu} \in \Gamma_{\vec{S}}]} \sum_{l \in \Omega_n} \mu_l, \quad \mu_{max,n}^{in} \triangleq \max_{[\vec{S}, \vec{\mu} \in \Gamma_{\vec{S}}]} \sum_{l \in \Theta_n} \mu_l.$$

We assume that each node n knows its own value of $\mu_{max,n}^{out}$, which is reasonable as nodes would typically have a pre-specified set of modulation and coding strategies to choose from, with a well-defined maximum. Assume that the flow control constants R_n^{max} are selected to satisfy $R_n^{max} \geq \mu_{max,n}^{out}$ for all n. As R_n^{max} is only used at node n, it can easily be set to satisfy this inequality. Assume utilities $g_n^{(c)}(r)$ are non-negative, non-decreasing, and concave, and define $G_{max} \triangleq \max_{[\sum_c r_n^{(c)} \leq R_n^{max} \forall n]} \sum_{n,c} g_n^{(c)}(r_n^{(c)})$. Define the constant B as follows:

$$B \triangleq \frac{1}{N} \sum_{n=1}^{N} \left[\left(R_n^{max} + \mu_{max,n}^{in}\right)^2 + \left(\mu_{max,n}^{out}\right)^2 \right]. \tag{11}$$

Theorem 2: If channel states are i.i.d. over timeslots and all active reservoirs have infinite backlog, then for any flow parameter $V > 0$ the CLC1 algorithm stabilizes the network and yields the following performance bounds:[2]

$$\overline{\sum_{n,c} U_n^{(c)}} \leq \frac{BN + VG_{max}}{2\mu_{sym}} \tag{12}$$

$$\liminf_{t \to \infty} \sum_{n,c} g_n^{(c)}\left(\overline{r}_n^{(c)}(t)\right) \geq \sum_{n,c} g_n^{(c)}\left(r_n^{*(c)}\right) - \frac{BN}{V} \tag{13}$$

[2] The algorithms developed in this paper yield similar results for general ergodic channel processes, where the B parameter in (12) and (13) is increased by an appropriate factor T related to steady state "mixing times" [1], [28].

where $(r_n^{*(c)})$ is the optimal solution of (2) subject to constraint (3), and where

$$\overline{\sum_{n,c} U_n^{(c)}} \triangleq \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\left[\sum_{n,c}\mathbb{E}\left\{U_n^{(c)}(\tau)\right\}\right]$$

$$\overline{r}_n^{(c)}(t) \triangleq \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{R_n^{(c)}(\tau)\right\} \qquad (14)$$

The above result holds for all $V > 0$. Thus, the value of V can be chosen so that BN/V is arbitrarily small, resulting in achieved utility that is arbitrarily close to optimal. This performance comes at the cost of a linear increase in network congestion with the parameter V. By Little's theorem [39], average queue backlog is proportional to average bit delay, and hence performance can be pushed towards optimality with a corresponding tradeoff in end-to-end network delay.

The proof of Theorem 2 follows from a novel Lyapunov drift argument, where the utility metric is incorporated into the drift condition so that stability and utility optimization can be simultaneously achieved. This analysis is provided in Section IV. Below we present a simple corollary that is useful in characterizing the performance of CLC1 under *suboptimal* resource allocations (proved at the end of Section IV).

Corollary 1: If the resource allocation policy of CLC1 is replaced with any (potentially randomized) resource allocation policy $\vec{\mu}(t)$ that satisfies the following for all slots t:

$$\sum_l W_l^*(t)\mathbb{E}\left\{\mu_l(t)|\underline{U}(t)\right\} \geq \theta\left(\max_{\vec{\mu}\in\Gamma_{\vec{S}(t)}}\sum_l W_l^*(t)\mu_l\right) - C$$

for some fixed constants θ and C such that $0 < \theta \leq 1$ and $C \geq 0$, then

$$\overline{\sum_{n,c} U_n^{(c)}} \leq \frac{2C + BN + VG_{max}}{2\mu_{sym}\theta} \qquad (15)$$

$$\liminf_{t\to\infty}\sum_{n,c} g_n^{(c)}\left(\overline{r}_n^{(c)}(t)\right) \geq \sum_{n,c} g_n^{(c)}\left(\tilde{r}_n^{*(c)}\right) - \frac{2C+BN}{V} \qquad (16)$$

where $(\tilde{r}_n^{*(c)})$ is the optimal solution to the following optimization:

$$\begin{aligned} \text{Maximize :} \quad & \sum_{n,c} g_n^{(c)}\left(r_n^{(c)}\right) \\ \text{Subject to :} \quad & \left(r_n^{(c)}\right) \in \theta\Lambda \\ & 0 \leq r_n^{(c)} \leq \lambda_n^{(c)}. \end{aligned} \qquad (17)$$

That is, allocating resources to come within a factor θ of the optimal solution of the CLC1 resource allocation yields a utility that is close to the optimal utility with respect to a θ scaled version of the capacity region. The above corollary is related to similar "sub-optimal" Lyapunov scheduling results presented for stability analysis (see, for example, [40], [41], and [1, Ch. 4.3.6]). It is also closely related to a similar "imperfect scheduling" result developed for utility optimization in [15] from a convex programming perspective.

E. Maximum Throughput and the Threshold Rule

Suppose utilities are linear, so that $g_n^{(c)}(r) = \alpha_n^{(c)} r$ for some non-negative weights $\alpha_n^{(c)}$. The resulting objective is to maximize the weighted sum of throughput, and the resulting FLOW algorithm has a simple threshold form, where some commodities receive as much of the R_n^{max} delivery rate as possible, while others receive none. In the special case where the user at node n desires communication with a single destination node c_n (so that $g_n^{(c)}(r) = 0$ for all $c \neq c_n$), the flow control algorithm (9) reduces to maximizing $V\alpha_n^{(c_n)}r - 2U_n^{(c_n)}r$ subject to $0 \leq r \leq R_n^{max}$, and the solution is the following threshold rule:

$$R_n^{(c_n)}(t) = \begin{cases} R_n^{max}, & \text{if } U_n^{(c_n)}(t) \leq \frac{V\alpha_n^{(c_n)}}{2} \\ 0, & \text{otherwise.}\end{cases}$$

The qualitative structure of this flow control rule is intuitive: When backlog in the source queue is large, we should refrain from sending new data. The simple threshold form is qualitatively similar to the threshold scheduling rule developed in [33] for server scheduling in a downlink with ON/OFF channels and deterministic constraints on the channel states and packet arrivals.

F. Proportional Fairness and the $1/U$ Rule

Consider now utility functions of the form $g_n^{(c)}(r) = \log(1 + \beta r_n^{(c)})$ (for some constant $\beta > 0$). It is shown in [12] that maximizing a sum of such utilities over any convex set Λ leads to *proportional fairness*.[3] In the special case when there is only one destination c_n for each user n, the flow control algorithm reduces to maximizing $V\log(1 + \beta r) - 2U_n^{(c_n)}r$ subject to $0 \leq r \leq R_n^{max}$, which leads to the following "$1/U$" flow control function:

$$R_n^{(c_n)}(t) = \min\left[\max\left[\frac{V}{2U_n^{(c_n)}(t)} - \frac{1}{\beta}, 0\right], R_n^{max}\right].$$

Here we see that the flow control valve restricts flow according to a continuous function of the backlog level at the source queue, being less conservative in its admission decisions when backlog is low and more conservative when backlog is high.

One drawback of this $1/U$ policy is that the resulting flow control variables $R_n^{(c)}(t)$ are real numbers (not necessarily integers or integer multiples of a given packet length), and hence it is implicitly assumed that packets can be fragmented for admission to the network. The CLC2 algorithm presented in Section V overcomes this issue.

IV. Performance Analysis

Here we prove Theorem 2. We first develop a novel Lyapunov drift result enabling stability and performance optimization to be performed using a single drift analysis.

A. Lyapunov Drift With Utility Metric

Let $\underline{U}(t) = (U_n^{(c)}(t))$ represent a process of queue backlogs that evolves according to some probability law, and define the *Lyapunov function* $L(\underline{U}) = \sum_{n,c}(U_n^{(c)})^2$. Let $R_n^{(c)}(t)$ repre-

[3] Strictly speaking, the proportionally fair allocation seeks to maximize $\sum_{n,c}\log(r_n^{(c)})$, leading to $\sum_{n,c}((\overline{r}_n^{*(c)} - r_n^{(c)})/\overline{r}_n^{*(c)}) \geq 0$ for any other operating point $(r_n^{(c)}) \in \Lambda$. We use non-negative utilities $\log(1 + \beta r)$ and thereby obtain a proportionally fair allocation with respect to the quantity $\overline{r}_n^{*(c)} + 1/\beta$, leading to $\sum_{n,c}((\overline{r}_n^{*(c)} - r_n^{(c)})/(\overline{r}_n^{*(c)} + 1/\beta)) \geq 0$. This can be used to approximate proportionally fair scheduling for large β. Alternatively, it can be used with $\beta = 1$, yielding a utility function $\log(1+r)$ which is different from proportionally fair utility but still has many desirable fairness properties.

sent an input process affecting the system, and suppose these values are bounded so that $\sum_{n,c} g_n^{(c)}(R_n^{(c)}(t)) \leq G_{max}$ for all t (for some value G_{max}). Assume utility functions $g_n^{(c)}(r)$ are non-negative and concave, and let g^* represent a "target utility" value. For a given queue backlog vector $\underline{U}(t)$, we define the *conditional Lyapunov drift* $\Delta(\underline{U}(t))$ as follows:

$$\Delta\left(\underline{U}(t)\right) \triangleq \mathbb{E}\left\{L\left(\underline{U}(t+1)\right) - L\left(\underline{U}(t)\right) | \underline{U}(t)\right\} \tag{18}$$

where the conditional expectation is with respect to the random one-step queueing dynamics given the current backlog $\underline{U}(t)$.

Lemma 1: (Lyapunov Optimization): If there are positive constants V, ϵ, B such that for all timeslots t and all unfinished work matrices $\underline{U}(t)$, the Lyapunov drift satisfies

$$\Delta\left(\underline{U}(t)\right) - V\sum_{n,c}\mathbb{E}\left\{g_n^{(c)}\left(R_n^{(c)}(t)\right)|\underline{U}(t)\right\} \leq B - \epsilon\sum_{n,c}U_n^{(c)}(t) - Vg^* \tag{19}$$

then time average utility and congestion satisfy

$$\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{n,c}\mathbb{E}\left\{U_n^{(c)}(\tau)\right\} \leq \frac{B+VG_{max}}{\epsilon} \tag{20}$$

$$\liminf_{t\to\infty}\sum_{n,c}g_n^{(c)}\left(\overline{r}_n^{(c)}(t)\right) \geq g^* - \frac{B}{V} \tag{21}$$

where $\overline{r}_n^{(c)}(t)$ is defined in (14).

Proof: Assume that (19) holds. Taking expectations over the distribution of $\underline{U}(t)$ and using the definition of $\Delta(\underline{U}(t))$ in (18) together with the law of iterated expectations yields

$$\mathbb{E}\left\{L\left(\underline{U}(t+1)\right) - L\left(\underline{U}(t)\right)\right\} - V\sum_{n,c}\mathbb{E}\left\{g_n^{(c)}\left(R_n^{(c)}(t)\right)\right\} \leq B - \epsilon\sum_{n,c}\mathbb{E}\left\{U_n^{(c)}(t)\right\} - Vg^*.$$

The above holds for all timeslots t. Summing over $t \in \{0, 1, \ldots, M-1\}$ yields

$$\mathbb{E}\left\{L\left(\underline{U}(M)\right)\right\} - \mathbb{E}\left\{L\left(\underline{U}(0)\right)\right\} - V\sum_{\tau=0}^{M-1}\sum_{n,c}\mathbb{E}\left\{g_n^{(c)}\left(R_n^{(c)}(\tau)\right)\right\} \leq BM - \epsilon\sum_{\tau=0}^{M-1}\sum_{n,c}\mathbb{E}\left\{U_n^{(c)}(\tau)\right\} - VMg^*. \tag{22}$$

Using non-negativity of the Lyapunov function and the utility functions as well as the fact that $\sum_{n,c} g_n^{(c)}(R_n^{(c)}(\tau)) \leq G_{max}$, we rearrange the terms of (22) and divide by $M\epsilon$ to yield

$$\frac{1}{M}\sum_{\tau=0}^{M-1}\sum_{n,c}\mathbb{E}\left\{U_n^{(c)}(\tau)\right\} - \frac{\mathbb{E}\left\{L\left(\underline{U}(0)\right)\right\}}{M\epsilon} \leq \frac{B+VG_{max}}{\epsilon.}$$

Taking the lim sup as $M \to \infty$ yields the backlog bound (20).

The utility bound (21) is proved similarly. Indeed, we again rearrange (22) and divide by MV to yield

$$\sum_{n,c}\frac{1}{M}\sum_{\tau=0}^{M-1}\mathbb{E}\left\{g_n^{(c)}\left(R_n^{(c)}(\tau)\right)\right\} \geq g^* - \frac{B+\mathbb{E}\left\{L\left(\underline{U}(0)\right)\right\}/M}{V}.$$

By concavity of $g_n^{(c)}(r)$ together with Jensen's inequality, it follows that

$$(1/M)\sum_{\tau=0}^{M-1}\mathbb{E}\{g_n^{(c)}(R_n^{(c)}(\tau))\} \leq g_n^{(c)}((1/M)\sum_{\tau=0}^{M-1}\mathbb{E}\{R_n^{(c)}(\tau)\})$$

Using this fact in the left hand side of the above inequality and taking the lim inf as $M \to \infty$ yields the result. □

The above lemma suggests that a good control strategy is to greedily minimize the following drift metric every timeslot:

$$\Delta\left(\underline{U}(t)\right) - V\sum_{n,c}\mathbb{E}\left\{g_n^{(c)}\left(R_n^{(c)}(t)\right)|\underline{U}(t)\right\}.$$

This is indeed the principle behind our control algorithm. To begin, we first develop an expression for Lyapunov drift from the queueing dynamics (8). First note that any general queue with backlog $U(t)$ and queueing law $U(t+1) = \max[U(t) - \mu(t), 0] + A(t)$ has a Lyapunov drift given by

$$\mathbb{E}\left\{U^2(t+1) - U^2(t)|U(t)\right\} \leq \mu_{max}^2 + A_{max}^2 - 2U(t)\mathbb{E}\left\{\mu(t) - A(t)|U(t)\right\} \tag{23}$$

where A_{max} and μ_{max} are upper bounds on the arrival and server variables $A(t)$ and $\mu(t)$. This well-known fact follows simply by squaring the queueing equation and taking expectations. Applying the general formula (23) to the specific queueing law (8) for queue (n, c) and summing the result over all (n, c) pairs yields the following expression for Lyapunov drift (see [1], [28] for details):

$$\Delta\left(\underline{U}(t)\right) \leq NB - 2\sum_{n,c}U_n^{(c)}(t) \times\mathbb{E}\left\{\sum_{l\in\Omega_n}\mu_l^{(c)}(t) - \sum_{l\in\Theta_n}\mu_l^{(c)}(t) - R_n^{(c)}(t)|\underline{U}(t)\right\} \tag{24}$$

where B is defined in (11).

Now define the *flow function* $\Psi(\underline{U}(t))$ and the *network function* $\Phi(\underline{U}(t))$ as follows:

$$\Psi\left(\underline{U}(t)\right) \triangleq \sum_{n,c}\mathbb{E}\left\{Vg_n^{(c)}\left(R_n^{(c)}\right) - 2U_n^{(c)}R_n^{(c)}|\underline{U}\right\} \tag{25}$$

$$\Phi\left(\underline{U}(t)\right) \triangleq 2\sum_{n,c}U_n^{(c)}\mathbb{E}\left\{\sum_{l\in\Omega_n}\mu_l^{(c)} - \sum_{l\in\Theta_n}\mu_l^{(c)}|\underline{U}\right\} \tag{26}$$

where we have represented $\underline{U}(t)$, $\mu_l^{(c)}(t)$, and $R_n^{(c)}(t)$ as $\underline{U}$, $\mu_l^{(c)}$, and $R_n^{(c)}$ for notational convenience. Subtracting the utility component $V\sum_{n,c}\mathbb{E}\{g_n^{(c)}(R_n^{(c)})|\underline{U}\}$ from both sides of (24) yields

$$\Delta\left(\underline{U}(t)\right) - V\sum_{n,c}\mathbb{E}\left\{g_n^{(c)}\left(R_n^{(c)}(t)\right)|\underline{U}\right\} \leq NB - \Phi\left(\underline{U}(t)\right) - \Psi\left(\underline{U}(t)\right). \tag{27}$$

Given a particular $\underline{U}(t)$ matrix at time t, the CLC1 policy is designed to *greedily minimize the right hand side of* (27) *over all possible routing, resource allocation, and flow control options*. We therefore have the following lemma:

Lemma 2: The $\Psi(\underline{U}(t))$ and $\Phi(\underline{U}(t))$ functions satisfy

$$\Psi^{CLC1}\left(\underline{U}(t)\right) \geq \sum_{n,c}\left[Vg_n^{(c)}\left(r_n^{*(c)}\right) - 2U_n^{(c)}(t)r_n^{*(c)}\right] \quad (28)$$

$$\Phi^{CLC1}\left(\underline{U}(t)\right) \geq 2\sum_{n,c}U_n^{(c)}(t)\mathbb{E}\left\{\sum_{l\in\Omega_n}\mu_l^{*(c)}(t) - \sum_{l\in\Theta_n}\mu_l^{*(c)}(t)|\underline{U}(t)\right\} \quad (29)$$

where $(r_n^{*(c)})$ are any particular values that satisfy (10) for all n, and $(\mu_l^{*(c)}(t))$ are any particular (potentially randomized) routing and resource allocation decisions at time t.

That the CLC1 flow control strategy (9) maximizes $\Psi(\underline{U}(t))$ over all feasible choices of the $R_n^{(c)}(t)$ values follows immediately by comparing (9) and (25). That the routing and resource allocation policy of CLC1 maximizes $\Phi(\underline{U}(t))$ is proven in [1], [28], and can be understood by switching the sums in the definition of $\Phi(\underline{U}(t))$:

$$\Phi\left(\underline{U}(t)\right) = 2\sum_l\sum_c\mathbb{E}\left\{\mu_l^{(c)}(t)|\underline{U}\right\}\left[U_{tran(l)}^{(c)} - U_{rec(l)}^{(c)}\right]. \quad (30)$$

Therefore, if $\Phi^{CLC1}(\underline{U}(t))$ represents the above function when the rates $\mu_l^{(c)}(t)$ are chosen at time t according to CLC1, and if $\Phi^*(\underline{U}(t))$ represents the above $\Phi(\underline{U}(t))$ function when any alternate feasible rates $\mu_l^{*(c)}(t)$ are chosen at time t (possibly via randomization), then from (30) we have

$$\Phi^*\left(\underline{U}(t)\right) \leq 2\sum_l\sum_c\mathbb{E}\left\{\mu_l^{*(c)}(t)|\underline{U}(t)\right\}W_l^*(t) \leq \Phi^{CLC1}\left(\underline{U}(t)\right) \quad (31)$$

where $W_l^*(t) = \max[U_{tran(l)}^{(c)}(t) - U_{rec(l)}^{(c)}(t), 0]$.

B. A Near-Optimal Operating Point

In order to use the Lyapunov drift result to establish the performance of the CLC1 algorithm, it is important to first compare performance to the utility of a *near-optimal* solution to the optimization problem (2)–(4). Specifically, for any $\epsilon > 0$, we define the set Λ_ϵ as follows:

$$\Lambda_\epsilon \triangleq \left\{\left(r_n^{(c)}\right) \mid \left(r_n^{(c)} + \epsilon 1_n^{(c)}\right) \in \Lambda, r_n^{(c)} \geq 0 \text{ for all } (n,c)\right\}$$

where $1_n^{(c)}$ is equal to 1 whenever $(n,c) \in \mathcal{D}$, and zero else. Thus, the set Λ_ϵ can be viewed as the resulting set of rate matrices within the network capacity region when an "ϵ-layer" of the boundary is stripped away from the D effective dimensions. Note that this set is compact and nonempty whenever $\epsilon \leq \mu_{sym}$ (where μ_{sym} is defined in Section III-A). The *near-optimal operating point* $(r_n^{*(c)}(\epsilon))$ is defined as a solution to the following optimization problem:[4]

$$\begin{aligned}\text{Maximize :} \quad & \sum_{n,c}g_n^{(c)}\left(r_n^{(c)}\right)\\ \text{Subject to :} \quad & \left(r_n^{(c)}\right) \in \Lambda_\epsilon\\ & r_n^{(c)} \leq \lambda_n^{(c)} \text{ for all } (n,c).\end{aligned} \quad (32)$$

This optimization differs from the optimization in (2)–(4) in that the set Λ is replaced by the set Λ_ϵ.

Lemma 3: (Continuity of Near-Optimal Solutions): If utility functions $g_n^{(c)}(r)$ are non-negative and concave, and if there is a positive scalar μ_{sym} such that $(\mu_{sym}) \in \Lambda$, then

$$\sum_{n,c}g_n^{(c)}\left(r_n^{*(c)}(\epsilon)\right) \to \sum_{n,c}g_n^{(c)}\left(r_n^{*(c)}\right) \quad \text{as } \epsilon \to 0. \quad (33)$$

Proof: The proof uses convexity of the capacity region Λ, and is given in [1, Ch. 5.5.2]. □

C. Derivation of Theorem 2

The proof of Theorem 2 relies on the following two inequalities:

$$\Psi^{CLC1}\left(\underline{U}(t)\right) \geq \sum_{n,c}\left[Vg_n^{(c)}\left(r_n^{*(c)}(\epsilon)\right) - 2U_n^{(c)}(t)r_n^{*(c)}(\epsilon)\right]$$

$$\Phi^{CLC1}\left(\underline{U}(t)\right) \geq 2\sum_{n,c}U_n^{(c)}(t)\left(r_n^{*(c)}(\epsilon) + \epsilon\right) \quad (34)$$

where $(r_n^{*(c)}(\epsilon))$ is the optimal solution of problem (32). The first inequality follows from (28) by using $r_n^{*(c)} = r_n^{*(c)}(\epsilon)$, noting that these are valid flow control choices because 1) all reservoirs are infinitely backlogged and so it is always possible to choose $R_n^{(c)}(t) = r_n^{*(c)}(\epsilon)$, and 2) the matrix $(r_n^{*(c)}(\epsilon))$ is within Λ and hence $\sum_c r_n^{*(c)}(\epsilon) \leq R_n^{max}$ for all n. Inequality (34) follows by plugging the particular $\mu_l^{*(c)}(t)$ policy of Theorem 1 (for $(r_n^{*(c)}(\epsilon) + \epsilon 1_n^{(c)}) \in \Lambda$) into (29).

Plugging the above two inequalities directly into the drift expression (27) yields the following for algorithm CLC1:

$$\begin{aligned}\Delta\left(\underline{U}(t)\right) - V\sum_{n,c}\mathbb{E}\left\{g_n^{(c)}\left(R_n^{(c)}(t)\right)|\underline{U}\right\} &\leq NB - 2\sum_{n,c}U_n^{(c)}(t)\left(r_n^{*(c)}(\epsilon) + \epsilon\right)\\ &\quad - V\sum_{n,c}g_n^{(c)}\left(r_n^{*(c)}(\epsilon)\right) + 2\sum_{n,c}U_n^{(c)}(t)r_n^{*(c)}(\epsilon).\end{aligned}$$

Canceling common terms yields

$$\Delta\left(\underline{U}(t)\right) - V\sum_{n,c}\mathbb{E}\left\{g_n^{(c)}\left(R_n^{(c)}(t)\right)|\underline{U}\right\} \leq NB - 2\epsilon\sum_{n,c}U_n^{(c)}(t) - V\sum_{n,c}g_n^{(c)}\left(r_n^{*(c)}(\epsilon)\right).$$

The above drift expression is in the exact form specified by Lemma 1. Thus, network congestion satisfies

$$\overline{\sum_{n,c}U_n^{(c)}} \leq (NB + VG_{max})/(2\epsilon) \quad (35)$$

and time average performance satisfies

$$\liminf_{t\to\infty}\sum_{n,c}g_n^{(c)}\left(\overline{r}_n^{(c)}(t)\right) \geq \sum_{n,c}g_n^{(c)}\left(r_n^{*(c)}(\epsilon)\right) - NB/V. \quad (36)$$

The performance bounds in (35) and (36) hold for any value ϵ such that $0 < \epsilon \leq \mu_{sym}$. However, the particular choice of ϵ only affects the bound calculation and does not affect the

[4] Note that the final constraint $(r_n^{(c)}) \leq (\lambda_n^{(c)})$ is satisfied automatically in the case of infinite traffic demand. We include the constraint here as this optimization is also important in the treatment of general traffic matrices $(\lambda_n^{(c)})$ in Section V.

CLC1 control policy or change any sample path of system dynamics. We can thus optimize the bounds separately over all possible ϵ values. The bound in (36) is clearly maximized by taking a limit as $\epsilon \to 0$, so that the right hand side converges to $\sum_{n,c} g_n^{(c)}(r_n^{*(c)}) - NB/V$ (using (33)). Conversely, the bound in (35) is minimized as $\epsilon \to \mu_{sym}$, yielding: $\overline{\sum_{n,c} U_n^{(c)}} \leq (NB + VG_{max})/(2\mu_{sym})$. This proves Theorem 2. □

The proof of Corollary 1 follows by noting that the sub-optimal allocation bound changes inequality (29) into

$$\Phi^{CLC1}\left(\underline{U}(t)\right) \geq -2C + 2\sum\nolimits_{n,c} U_n^{(c)}(t) \times \mathbb{E}\left\{\sum\nolimits_{l\in\Omega_n} \theta\mu_l^{*(c)}(t) - \sum\nolimits_{l\in\Theta_n} \theta\mu_l^{*(c)}(t)|\underline{U}(t)\right\}$$

which thus changes inequality (34) into

$$\Phi^{CLC1}\left(\underline{U}(t)\right) \geq -2C + 2\theta\sum\nolimits_{n,c} U_n^{(c)}(t)\left(r_n^{*(c)}(\epsilon) + \epsilon\right).$$

The proof then proceeds exactly as before, as noted in [1]. □

V. Scheduling With Arbitrary Input Rates

The algorithm CLC1 assumes there is always an amount of data $R_n^{(c)}(t)$ available in reservoir (n,c), where the flow variable $R_n^{(c)}(t)$ is chosen only with respect to the R_n^{max} constraint. Here we assume that all transport layer storage reservoirs have a finite (possibly zero) buffer, and let $L_n^{(c)}(t)$ represent the current backlog in the reservoir buffer. The flow control decisions are now subject to the additional constraint $R_n^{(c)}(t) \leq L_n^{(c)}(t) + A_n^{(c)}(t)$ (where $A_n^{(c)}(t)$ is the amount of new commodity c data exogenously arriving to node n at slot t). Any arriving data that is not immediately admitted to the network is stored in the reservoir, or dropped if the reservoir has no extra space.

Assume the $A_n^{(c)}(t)$ arrivals are i.i.d. over timeslots with arrival rates $\lambda_n^{(c)} = \mathbb{E}\{A_n^{(c)}(t)\}$. It can be shown that for any matrix $(\lambda_n^{(c)})$ (possibly outside of the capacity region), modifying the CLC1 flow algorithm to maximize (9) subject to the additional reservoir backlog constraint yields the same performance guarantees (12) and (13) *when utility functions are linear* [1]. For nonlinear utilities, such a strategy can be shown to maximize the time average of $\sum_{n,c}\mathbb{E}\{g_n^{(c)}(R_n^{(c)}(t))\}$ over all strategies that make immediate admission/rejection decisions upon arrival, but may not necessarily maximize $\sum_{n,c} g_n^{(c)}(\mathbb{E}\{R_n^{(c)}(t)\})$, which is the utility metric of interest. We solve this problem with a novel technique of defining additional *flow state variables* $Z_n^{(c)}(t)$. The result can be viewed as a general framework for stochastic network optimization.

Define *flow state variables* $Z_n^{(c)}(t)$ for each reservoir (n,c), and fix $Z_n^{(c)}(0) = VR_n^{max}/2$ for all (n,c) (any initial condition can be used, this one works well experimentally). For each flow control process $R_n^{(c)}(t)$, we define a new process $Y_n^{(c)}(t)$ as follows:

$$Y_n^{(c)}(t) \triangleq R_n^{max} - R_n^{(c)}(t) \tag{37}$$

and note that $Y_n^{(c)}(t) \geq 0$ for all t. The $Y_n^{(c)}(t)$ variables represent the difference between the maximum value and the actual value of admitted data on session (n,c). The $Z_n^{(c)}(t)$ state variables are updated every slot according to the following "queue-like" iteration:

$$Z_n^{(c)}(t+1) = \max\left[Z_n^{(c)}(t) - \gamma_n^{(c)}(t), 0\right] + Y_n^{(c)}(t) \tag{38}$$

where $\{\gamma_n^{(c)}(t)\}$ are additional flow control decision variables. Define the "cost" function:

$$h_n^{(c)}(\gamma) \triangleq g_n^{(c)}\left(R_n^{max}\right) - g_n^{(c)}\left(R_n^{max} - \gamma\right). \tag{39}$$

Let $\overline{\gamma}_n^{(c)}$ represent the time average value of the decision variables $\gamma_n^{(c)}(t)$. We design a policy to stabilize the network queues $U_n^{(c)}(t)$ *and* the flow state 'queues' $Z_n^{(c)}(t)$ while minimizing the cost $\sum_{n,c} h_n^{(c)}(\overline{\gamma}_n^{(c)})$. The intuitive interpretation of this goal is as follows: If the $Z_n^{(c)}(t)$ queues are stabilized, it must be the case that the time average of the 'server process' $\gamma_n^{(c)}(t)$ is greater than or equal to the time average of the 'arrival process' $Y_n^{(c)}(t) : \overline{Y}_n^{(c)} \leq \overline{\gamma}_n^{(c)}$. From (37), this implies $\overline{r}_n^{(c)} \geq R_n^{max} - \overline{\gamma}_n^{(c)}$, and hence

$$\sum_{n,c} h_n^{(c)}\left(\overline{\gamma}_n^{(c)}\right) = \sum_{n,c} g_n^{(c)}\left(R_n^{max}\right) - \sum_{n,c} g_n^{(c)}\left(R_n^{max} - \overline{\gamma}_n^{(c)}\right) \geq \sum_{n,c} g_n^{(c)}\left(R_n^{max}\right) - \sum_{n,c} g_n^{(c)}\left(\overline{r}_n^{(c)}\right).$$

Thus, minimizing $\sum_{n,c} h_n^{(c)}(\overline{\gamma}_n^{(c)})$ over all feasible $\overline{\gamma}_n^{(c)}$ values is intimately related to maximizing $\sum_{n,c} g_n^{(c)}(\overline{r}_n^{(c)})$ over all feasible $\overline{r}_n^{(c)}$ values.

Cross Layer Control Policy 2 (CLC2): Let η be any fixed constant such that $0 < \eta \leq 1$. Every timeslot and for each node n, choose $R_n^{(c)}(t) = x_n^{(c)}$ to solve

$$\begin{aligned} \text{Maximize}: \quad & \sum_c \left[\eta Z_n^{(c)}(t) - U_n^{(c)}(t)\right] x_n^{(c)} \\ \text{Subject to}: \quad & \sum_c x_n^{(c)} \leq R_n^{max} \\ & x_n^{(c)} \leq L_n^{(c)}(t) + A_n^{(c)}(t). \end{aligned} \tag{40}$$

Additionally, the flow controllers at each node n choose $\gamma_n^{(c)}(t)$ for each session (n,c) to solve

$$\begin{aligned} \text{Maximize}: \quad & Vg_n^{(c)}\left(R_n^{max} - \gamma_n^{(c)}\right) + 2\eta Z_n^{(c)}(t)\gamma_n^{(c)} \\ \text{Subject to}: \quad & 0 \leq \gamma_n^{(c)} \leq R_n^{max}. \end{aligned} \tag{41}$$

The flow states $Z_n^{(c)}(t)$ are then updated according to (38). Routing and resource allocation within the network is the same as in CLC1.

The optimization of $R_n^{(c)}(t)$ in (40) is solved by a simple 'bang-bang' control policy, where no data is admitted from reservoir (n,c) if $U_n^{(c)}(t) > \eta Z_n^{(c)}(t)$, and otherwise as much data as possible is delivered from the commodities of node n with the largest non-negative values of $[\eta Z_n^{(c)}(t) - U_n^{(c)}(t)]$, subject to the R_n^{max} constraint. These bang-bang decisions also enable the strategy to be implemented optimally in systems where admitted data is constrained to integral units, a feature that CLC1 does not have. As in CLC1, the constraint

$\sum_c x_n^{(c)} \leq R_n^{max}$ can be replaced by the simpler constraint $x_n^{(c)} \leq R_n^{max}$, at the cost of increasing the system delay bound.

The $\gamma_n^{(c)}(t)$ variable assignment in (41) involves maximizing a concave function of a single variable, and can be solved easily by finding the critical points over $0 \leq \gamma_n^{(c)} \leq R_n^{max}$. For example, if $g_n^{(c)}(r) = \log(1 + \beta r)$, it can be shown that

$$\gamma_n^{(c)} = \min\left[\max\left[\frac{1}{\beta} + R_n^{max} - \frac{V}{2\eta Z_n^{(c)}(t)}, 0\right], R_n^{max}\right].$$

Suppose channels and arrivals are i.i.d. over timeslots, and let $\lambda_n^{(c)} = \mathbb{E}\{A_n^{(c)}(t)\}$. We assume that $\lambda_n^{(c)} = 0$ whenever $(n, c) \notin \mathcal{D}$. For simplicity of exposition, we further assume that new arrivals to node n are deterministically bounded by R_n^{max}, so that $\sum_c A_n^{(c)}(t) \leq R_n^{max}$ every slot.

Theorem 3: For arbitrary rate matrices $(\lambda_n^{(c)})$ (possibly outside of the capacity region), for any $V > 0$, and for any reservoir buffer size (possibly zero), the CLC2 algorithm stabilizes the network and yields a congestion bound:

$$\overline{\sum_{n,c} U_n^{(c)}} \leq \frac{NB + 2\eta N \sum_n (R_n^{max})^2 + VG_{max}}{2\mu_{sym}}$$

Further, the time average utility satisfies

$$\liminf_{t\to\infty} \sum_{n,c} g_n^{(c)}\left(\overline{r}_n^{(c)}(t)\right) \geq \sum_{n,c} g_n^{(c)}\left(r_n^{*(c)}\right) - \frac{NB + 2\eta N \sum_n (R_n^{max})^2}{V}.$$

Proof: Define the Lyapunov function $L(\underline{U}, \underline{Z}) = \sum_{n,c}(U_n^{(c)})^2 + \eta \sum_{n,c}(Z_n^{(c)})^2$. The drift expression for this function is given by summing the drift of the $U_n^{(c)}(t)$ queues and the $Z_n^{(c)}(t)$ queues using the general formula (23), where the queueing laws are given by (8) and (38):

$$\begin{aligned}\Delta(\underline{U}(t), \underline{Z}(t)) &+ \sum_{n,c} V\mathbb{E}\left\{h_n^{(c)}\left(\gamma_n^{(c)}(t)\right)|\underline{U}, \underline{Z}\right\}\\ &\leq NB + 2\eta N \sum_n (R_n^{max})^2 - \Phi(\underline{U}(t))\\ &\quad + 2\sum_{n,c} \mathbb{E}\left\{U_n^{(c)}(t)R_n^{(c)}(t) + \eta Y_n^{(c)}(t)Z_n^{(c)}(t)|\underline{U}, \underline{Z}\right\}\\ &\quad - \sum_{n,c} \mathbb{E}\left\{2\eta Z_n^{(c)}(t)\gamma_n^{(c)}(t) - Vh_n^{(c)}\left(\gamma_n^{(c)}(t)\right)|\underline{U}, \underline{Z}\right\} \qquad (42)\end{aligned}$$

where we have added to both sides of the above expression the cost term $\sum_{n,c} V\mathbb{E}\{h_n^{(c)}(\gamma_n^{(c)}(t))|\underline{U}, \underline{Z}\}$. The CLC2 policy is designed *to minimize the right hand side of the above expression over all possible policies*. Thus, similar to the proof of CLC1, plugging particular policies gives bounds for which the Lyapunov Optimization Lemma (Lemma 1) can be applied, proving stability of all queues, proving that $\mathbb{E}\{Z_n^{(c)}(t)\}/t \to 0$, and leading to the result (see [2] for details). □

We note that the transport layer storage reservoir does not impact the above result, and hence a size-zero reservoir (in which all data is immediately accepted or dropped upon arrival) yields the same performance guarantees. We further note that it is possible to use the same proof to show that if the input rate matrix $(\lambda_n^{(c)})$ is in the relative interior of the capacity region Λ (so that $(\lambda_n^{(c)} + \delta 1_n^{(c)}) \in \Lambda$ for some $\delta > 0$), then the same utility bound of Theorem 3 holds, but the congestion can be bounded by a constant that depends on the proximity to the boundary of the capacity region, but does not depend on V (we omit these details for brevity). This is intuitively satisfying, as we know from the results of [1], [28] that the DRPC algorithm (without flow control) bounds average congestion by a similar constant whenever input rates are interior to the capacity region.

VI. Simulation Results

Here we simulate the CLC2 policy for three simple network examples. We assume throughout that $\eta = 1/N$. We begin with the 2-queue downlink example of Section II. Packets arrive from each stream according to Bernoulli processes, and we assume there are no transport layer storage reservoirs, so that all arriving data is either immediately admitted or dropped.[5] As before, we assume channel probabilities are given by $p_1 = 0.5$, $p_2 = 0.6$, and consider one hundred different rate pairs (λ_1, λ_2) that are linearly scaled towards the point (0.5, 1.0). For each point we simulate the CLC2 algorithm for 3 million timeslots, using $V = 10000$, $R_n^{max} = 2$, and $g_1(r) = g_2(r) = \log(1 + r)$. Note that in this case, we have $\mu_{max}^{in} = 0$, $\mu_{max}^{out} = 1$, so by (11) we have $B = 5$. Thus, for $V = 10000$ we are ensured by Theorem 3 that the resulting utility associated with each rate vector (λ_1, λ_2) differs from the optimum utility by no more than $(5 + 8)/V = 0.0013$ (note that $N = 1$ for this simple example, as there is only 1 transmitting node). The simulation results are shown in Fig. 3(a), where the achieved throughput increases to the capacity boundary and then moves directly to the fair point (0.4, 0.4).

In Fig. 3(b) we treat the same situation with the exception that utility for user 2 is modified to $1.28\log(1 + r)$. This illustrates the ability to handle *priority service*, as user 2 traffic is now given favored treatment. From the figure, we see that as input rates are increased the resulting throughput reaches the capacity boundary and then *moves in a new direction*, settling and remaining on the new optimal operating point (0.23, 0.57) once input rates dominate this point.

Note that for this example, we have $\mu_{sym} = 0.4$ and $G_{max} = 0.784$. Thus, by Theorem 3, we know

$$\overline{U}_1 + \overline{U}_2 \leq \frac{13 + 0.784V}{0.8}. \qquad (43)$$

The above bound holds for any input rate vector (λ_1, λ_2), including vectors that are far outside of the capacity region. We next keep the same utility as in Fig. 3(b) but fix the input rate to $(\lambda_1, \lambda_2) = (0.5, 1.0)$, which dominates the optimal operating point (0.23, 0.57) (so that utility optimal control should achieve this point). In Fig. 3(c) we plot the resulting average queue congestion as V is varied from 1 to 10^4, together with the bound (43). As suggested by the bound, the delay grows linearly with V. In Fig. 3(d) we see how the achieved throughput of CLC2 approaches the optimal operating point (0.23, 0.57) as V is increased.

A. Packet Switches

Here we consider a simple 3×3 packet switch with a crossbar switch fabric [23], [26]. Packets arrive from three different input ports, and each packet is destined for one of three output ports.

[5] Simulations for infinite transport reservoirs yield almost identical results.

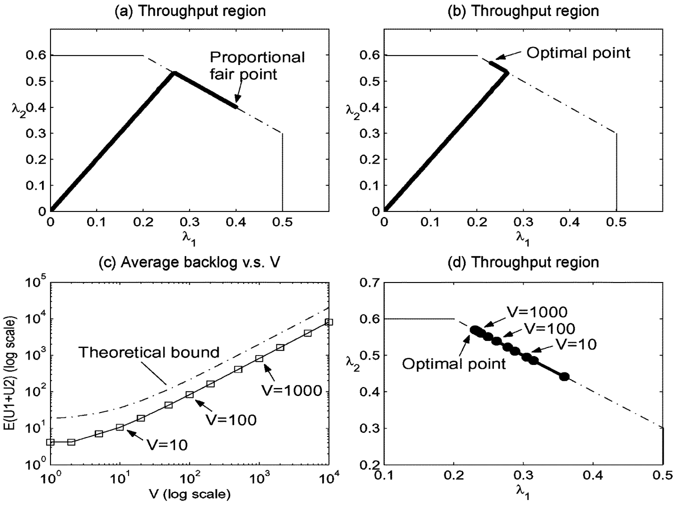

Fig. 3. Simulation of CLC2. (a) Linearly increasing (λ_1, λ_2) to (0.5, 1.0) for $V = 10000$ and $g_1(r) = g_2(r) = \log(1 + r)$. (b) Modifying utility 2 to: $g_2(r) = 1.28 \log(1+r)$. (c)-(d) Fixing $(\lambda_1, \lambda_2) = (0.5, 1.0)$ and illustrating delay and throughput versus V.

Rates (λ_{ij})		
.45	.1	.4
.1	.7	.15
.4	.15	.4

Throughput (r_{ij})		
.450	.100	.399
.100	.695	.148
.399	.149	.400

Backlog ($\overline{U}_{ij}$)		
3.3	2.4	3.6
2.4	2.9	2.7
3.6	2.7	3.4

(a)

Rates (λ_{ij})		
.9	.2	.3
0	.4	.2
0	.5	0

Throughput (r_{ij})		
.598	.100	.298
0	.399	.200
0	.500	0

Backlog ($\overline{U}_{ij}$)		
31.6	45.3	32.1
0	14.1	.29
0	14.2	0

(b)

Fig. 4. Simulation results for the CLC2 algorithm with $V = 100$ and zero reservoir buffers. Simulations were run over four million timeslots. (a) Simulation of a switch with feasible traffic. (b) Simulation of an overloaded switch.

We let λ_{ij} represent the rate of packets arriving to input i and destined for output j. All packets are stored in *virtual input queues* according to their destinations, and we let $U_{ij}(t)$ represent the current number of backlogged packets waiting at input i to be delivered to output j. The system is timeslotted, and the crossbar fabric limits scheduling decisions to *permutation matrices*, where no input can transfer more than one packet per slot, and no output can receive more than one packet per slot. Thus, the following *feasibility constraints* are required for stability:

$$\sum_{i=1}^{3} \lambda_{ij} \leq 1, \; \forall j \in \{1,2,3\}, \quad \sum_{j=1}^{3} \lambda_{ij} \leq 1, \; \forall i \in \{1,2,3\}.$$

We consider i.i.d. Bernoulli arrivals, and apply the CLC2 algorithm using $\log(1 + r)$ utility functions, $V = 100$, and $R_{max} = 3$ (the maximum number of arrivals to a single input during a slot). In this example we assume that all reservoirs have zero buffers, so that admission/rejection decisions must be made immediately upon packet arrival. Resource allocation for CLC2 in this context is equivalent to Maximum Weight Matching. In Fig. 4(a) we present simulation results for a case when the sum rate to every input port and every output port is exactly 0.95. Note that average queue backlog is kept very low, while the resulting throughput is almost identical to the input rate. This illustrates that the CLC2 algorithm accepts almost all packets into the system, and accomplishes this without a-priori knowledge that the input traffic is feasible.

We next apply the same CLC2 algorithm to a switch where input port 1 and output port 2 are overloaded, as shown in Fig. 4(b). The resulting throughput from the simulation is given in the figure, and is almost indistinguishable from the utility maximizing solution of the optimization problem (2)–(4). The average backlog in each queue is no more than 45.3 packets.

B. Heterogeneous Multi-Hop Networks

Here we consider the multi-hop network of Fig. 1, consisting of wireless sensor nodes (nodes {6,7,8,9}), a wireline network, and a wireless basestation that transmits to two mobile users. All packets have fixed lengths. The wireline links are bidirectional and can transmit 3 packets in each direction during a single slot. The basestation node 0 can transmit to only one mobile user per slot, and the downlinks to each user independently vary between ON and OFF states according to Bernoulli processes, with equal likelihood of being ON or OFF. The wireless links of the sensor network are always ON, and can support one packet transfer per slot. However, due to interference between the various sensor nodes, we assume that only one sensor link can be activated per slot (including the outgoing wireless links of the access nodes 4 and 5).

We assume there are four independent sessions using the network: Two sessions originate from node 9 and consist of packets destined for nodes 3 and 1, and two sessions originate from node 4 and consist of packets destined for nodes 8 and 2. All arrival processes are i.i.d. and Bernoulli, with arrival rates $\lambda_{93} = \lambda_{91} = \lambda_{48} = \lambda_{42} = 0.7$.

These arrival rates are not supportable by the network. Indeed, note that all packets originating at node 9 must travel over at least 2 sensor links before reaching a wireline access node. Likewise, all data from the λ_{48} stream requires at least two hops through the sensor network. Because at most one sensor link can be activated on any timeslot, it follows that $2r_{93} + 2r_{91} + 2r_{48} \leq 1$ is a necessary condition for network stability, where r_{ij} is the throughput of (i, j) traffic. The basestation places the following additional constraints on the network flows: $r_{91} \leq 1/2$, $r_{42} \leq 1/2$, and $r_{91} + r_{42} \leq 3/4$. It is not difficult to verify that these necessary conditions describe the feasible flows for the network, as the wired links do not impose further constraints. Assuming that all sessions have identical utility functions $g_{ij}(r) = \log(1 + r)$, the optimally fair flows are thus given by $r^*_{91} = r^*_{93} = r^*_{48} = 1/6 = 0.1667$, $r^*_{42} = 0.5$.

We implement CLC2 for this network, using $V = 1000$, $R_{max} = 2$, and assuming infinite buffer reservoirs. Note that sensor link activations are determined every slot by choosing the link with the largest differential backlog. The resulting average queue length, summed over all queues in the system, is 858.9 packets. The throughputs are: $r_{91} = 0.1658$, $r_{93} = 0.1662$, $r_{48} = 0.1678$, $r_{42} = 0.5000$. We note that this performance is not possible unless almost all packets take their optimal 2-hop paths through the sensor network, and hence the backpressure routing learns the optimal routes.

VII. Distributed Control for Wireless Networks

Theorem 2 and its corollary demonstrate that it is important to allocate resources in an effort to maximize $\sum_l W_l^*(t)\mu_l$, even if implementation complexity precludes realization of the full maximum (this is also highlighted in [1, Ch. 4.3.6]). Indeed, it is often difficult to achieve the full maximum, especially in distributed networks with interference. However, there are several basic network models for which it is possible to come within a constant factor of optimality using simplified scheduling techniques, and this is an area of recent active interest [15], [42], [43].

For example, consider a wireless network where every node transmits over an orthogonal frequency band. Assume nodes can transmit or receive from at most one link at a time, and nodes cannot simultaneously transmit and receive. Such a model is recently considered in [15], [42]. Let $b_l(t)$ represent the rate that link l can transmit data during slot t if it is chosen for transmission. The problem of maximizing $\sum_l W_l^*(t)\mu_l$ in this case amounts to finding a *maximum weight match* (MWM) in the underlying network graph. Such matchings are important for network stability problems (see [20], [23]), although they are difficult to compute in a distributed manner because the matching constraints couple all decisions throughout the network. However, it is well known that a simple contention based algorithm for greedily selecting a matching comes within a factor of two of the optimal solution.

Specifically, the contention scheme goes through several rounds before finalizing a set of links to activate. In the first round, each node attempts to activate its largest weight outgoing link. Any resulting contentions are resolved locally by (tentatively) selecting the links with the largest weight among all competitors, breaking ties arbitrarily. The next round proceeds by having each non-selected node attempt to activate its largest weight link that does not contend with any selected link of equal or larger weight. Contentions are again resolved locally, placing the largest weight contenders in the "selected" group and removing any lesser weight contenders previously selected. Proceeding this way, it is easy to see that each round of contention permanently adds at least one new link, and so after at most N rounds the algorithm converges to a matching in which all non-active links are adjacent to at least one active link with equal or larger weight. It can be shown that this matching is within a factor of 2 of optimality, so that the condition of Corollary 1 holds with $\theta = 1/2$.

A more general interference model is recently considered in [43], where each link l has a set of K_l "competitor links" that cannot be activated if link l is transmitting. Define $K_{max} \overset{\Delta}{=} \max_l K_l$. It can be shown that, within N rounds, a similar greedy contention scheme finds a feasible set of links to activate that comes within a factor of K_{max} of the optimal solution (see [43] for a similar result). A simpler *random access* scheme that uses a collision model and requires only one contention round is as follows: Each link independently attempts activation with probability $1/K_{max}$. If a link has no contenders, then it is successfully activated. Else, it remains idle. This random access scheme is similar to the distributed random access scheme specified in [1], [28]. It is clear that this scheme satisfies

$$\begin{aligned}&\sum\nolimits_l W_l^*(t)\mathbb{E}\left\{\mu_l(t)|\underline{U}(t)\right\}\\&\quad\geq \sum\nolimits_l W_l^*(t)\mathbb{E}\left\{b_l(t)|\underline{U}(t)\right\}\left(\frac{1}{K_{max}}\right)\left(1-\frac{1}{K_{max}}\right)^{K_{max}}\end{aligned}$$

and hence the left hand side is within a factor of $1/K_{max}(1 - 1/K_{max})^{K_{max}}$ of the sum of weights over all links in the network. Hence, the expectation is certainly within this same factor of the optimal solution (which must conform to valid link activation sets). Note that $1/K_{max}(1-1/K_{max})^{K_{max}} \approx 1/(K_{max}e)$ (for large K_{max}), and hence the factor $1/e$ can roughly be viewed as the cost of using this single-round random access scheme in comparison to the multi-round greedy approach. The solution can further be improved by using multiple random access rounds and/or by restricting attempts that lead to obvious contentions (such as attempting to activate two outgoing links from the same node).

VIII. Conclusion

We have presented a fundamental approach to stochastic network control for heterogeneous data networks. Simple strategies were developed that perform arbitrarily close to the optimally fair throughput point (regardless of the input traffic matrix), with a corresponding tradeoff in end-to-end network delay. The strategies involve resource allocation and routing decisions that are decoupled over the independent portions of the network, and flow control algorithms that are decoupled over independent control valves at every node. Flow controllers require knowledge only of the queue backlog in their respective source nodes. We note that this technique of implementing flow control only at the sources is crucial to ensure no network resources are wasted transmitting data that will eventually be dropped. It is remarkable that the overall strategy does not require knowledge of input rates, channel statistics, or the global network topology. Although i.i.d. assumptions were made to simplify exposition, the same policies can be shown to offer similar performance (with modified delay expressions) for arbitrary ergodic arrivals and channels, and are robust to cases when channel probabilities or arrival rates change over time [1], [4]. We believe that such theory-driven networking strategies will impact the design and operation of future data networks.

References

[1] M. J. Neely, "Dynamic power allocation and routing for satellite and wireless networks with time varying channels," Ph.D. dissertation, LIDS, Massachusetts Inst. Technol., Cambridge, MA, 2003.

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

[3] M. J. Neely, "Energy optimal control for time varying wireless networks," in *Proc. IEEE INFOCOM*, Mar. 2005.

[4] 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.

[5] J. W. Lee, R. R. Mazumdar, and N. B. Shroff, "Downlink power allocation for multi-class CDMA wireless networks," in *Proc. IEEE INFOCOM*, 2002.

[6] R. Berry, P. Liu, and M. Honig, "Design and analysis of downlink utility-based schedulers," in *Proc. 40th Allerton Conf. Communication, Control, and Computing*, Oct. 2002.

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

[8] M. Chiang, "Balancing transport and physical layer in wireless multihop networks: Jointly optimal congestion control and power control," *IEEE J. Sel. Area Commun.*, vol. 1, no. 23, pp. 104–116, Jan. 2005.

[9] L. Xiao, M. Johansson, and S. Boyd, "Simultaneous routing and resource allocation for wireless networks," in *Proc. 39th Allerton Conf. Communication, Control, and Computing*, Oct. 2001.

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

[11] P. Marbach, "Priority service and max-min fairness," in *Proc. IEEE INFOCOM*, 2002.

[12] F. Kelly, "Charging and rate control for elastic traffic," *Eur. Trans. Telecommun.*, vol. 8, pp. 33–37, 1997.

[13] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: Shadow prices, proportional fairness, and stability," *J. Oper. Res. Soc.*, vol. 49, pp. 237–252, 1998.

[14] S. H. Low and D. E. Lapsley, "Optimization flow control, i: Basic algorithm and convergence," *IEEE/ACM Trans. Networking*, vol. 7, no. 6, pp. 861–875, Dec. 1999.

[15] X. Lin and N. B. Shroff, "The impact of imperfect scheduling on cross-layer rate control in wireless networks," in *Proc. IEEE INFOCOM*, 2005.

[16] B. A. Movsichoff, C. M. Lagoa, and H. Che, "Decentralized optimal traffic engineering in connectionless networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 293–303, Feb. 2005.

[17] S. H. Low, "A duality model of TCP and queue management algorithms," *IEEE Trans. Networking*, vol. 11, no. 4, Aug. 2003.

[18] X. Liu, E. K. P. Chong, and N. B. Shroff, "A framework for opportunistic scheduling in wireless networks," *Comput. Netw.*, vol. 41, no. 4, pp. 451–474, Mar. 2003.

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

[20] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Contr.*, vol. 37, no. 12, pp. 1936–1949, Dec. 1992.

[21] L. Tassiulas and A. Ephremides, "Dynamic server allocation to parallel queues with randomly varying connectivity," *IEEE Trans. Inf. Theory*, vol. 39, pp. 466–478, Mar. 1993.

[22] P. R. Kumar and S. P. Meyn, "Stability of queueing networks and scheduling policies," *IEEE Trans. Autom. Contr.*, vol. 40, no. 2, pp. 251–260, Feb. 1995.

[23] N. McKeown, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," in *IEEE INFOCOM*, 1996.

[24] N. Kahale and P. E. Wright, "Dynamic global packet routing in wireless networks," in *Proc. IEEE INFOCOM*, 1997.

[25] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, and P. Whiting, "Providing quality of service over a shared wireless link," *IEEE Commun. Mag.*, vol. 39, no. 2, pp. 150–154, Feb. 2001.

[26] E. Leonardi, M. Mellia, F. Neri, and M. A. Marsan, "Bounds on average delays and queue size averages and variances in input-queued cell-based switches," in *Proc. IEEE INFOCOM*, 2001.

[27] M. J. Neely, E. Modiano, and C. E. Rohrs, "Power allocation and routing in multi-beam satellites with time varying channels," *IEEE Trans. Networking*, vol. 11, no. 1, pp. 138–152, Feb. 2003.

[28] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic power allocation and routing for time varying wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 89–103, Jan. 2005.

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

[30] S. Borst, "User-level performance of channel-aware scheduling algorithms in wireless data networks," in *Proc. IEEE INFOCOM*, 2003.

[31] D. N. Tse, "Optimal power allocation over parallel broadcast channels," in *Proc. Int. Symp. Information Theory (ISIT)*, Jun. 1997.

[32] H. Kushner and P. Whiting, "Asymptotic properties of proportional-fair sharing algorithms," in *Proc. 40th Allerton Conf. Communication, Control, and Computing*, 2002.

[33] V. Tsibonis, L. Georgiadis, and L. Tassiulas, "Exploiting wireless channel state information for throughput maximization," in *Proc. IEEE INFOCOM*, Apr. 2003.

[34] L. Li and A. Goldsmith, "Capacity and optimal resource allocation for fading broadcast channels: Part i: Ergodic capacity," *IEEE Trans. Inf. Theory*, pp. 1083–1102, Mar. 2001.

[35] A. Stolyar, "Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm," *Queueing Syst.*, vol. 50, pp. 401–457, 2005.

[36] J. W. Lee, R. R. Mazumdar, and N. B. Shroff, "Opportunistic power scheduling for dynamic multiserver wireless systems," *IEEE Trans. Wireless Commun.*, vol. 5, no. 6, pp. 1506–1515, Jun. 2006.

[37] P. Kall and S. W. Wallace*, Stochastic Programming*. New York: Wiley, 1994.

[38] M. J. Neely, "Super-fast delay tradeoffs for utility optimal fair scheduling in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1489–1501, Aug. 2006.

[39] D. P. Bertsekas and R. Gallager*, Data Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[40] D. Shah and M. Kopikare, "Delay bounds for approximate maximum weight matching algorithms for input queued switches," in *Proc. IEEE INFOCOM*, Jun. 2002.

[41] M. J. Neely, E. Modiano, and C. E. Rohrs, "Tradeoffs in delay guarantees and computation complexity for $n \times n$ packet switches," in *Proc. Conf. Inf. Sci. Syst. (CISS)*, Princeton, NJ, Mar. 2002.

[42] X. Wu and R. Srikant, "Regulated maximal matching: A distributed scheduling algorithm for multi-hop wireless networks with node-exclusive spectrum sharing," in *Proc. IEEE Conf. Decision Contr.*, 2005.

[43] P. Chaporkar, K. Kar, and S. Sarkar, "Throughput guarantees through maximal scheduling in wireless networks," in *Proc. 43rd Allerton Conf. Communication, Control and Computing*, Sep. 2005.

Michael J. Neely (S'00–M'02) received the B.S. degrees in both electrical engineering and mathematics from the University of Maryland, College Park, in 1997. He was then awarded a three-year Department of Defense NDSEG Fellowship for graduate study at the Massachusetts Institute of Technology, Cambridge, where he received the M.S. degree in 1999 and the Ph.D. degree in 2003, both in electrical engineering.

In 2004, he joined the faculty of the Electrical Engineering Department, University of Southern California, Los Angeles, where he is currently an Assistant Professor. His research is in the area of stochastic network optimization for wireless and ad hoc networks.

Eytan Modiano (S'90–M'93–SM'00) received the B.S. degree in electrical engineering and computer science from the University of Connecticut at Storrs in 1986 and the M.S. and Ph.D. degrees, both in electrical engineering, from the University of Maryland, College Park, MD, in 1989 and 1992, respectively.

He was a Naval Research Laboratory Fellow between 1987 and 1992, and a National Research Council Post Doctoral Fellow during 1992–1993. Between 1993 and 1999, he was with MIT Lincoln Laboratory where he was the project leader for MIT Lincoln Laboratory's Next Generation Internet (NGI) project. Since 1999, he has been on the faculty at MIT, where he is presently an Associate Professor. His research is on communication networks and protocols with emphasis on satellite, wireless, and optical networks.

Dr. Modiano is currently an Associate Editor for IEEE TRANSACTIONS ON INFORMATION THEORY, *The International Journal of Satellite Communications*, and IEEE/ACM TRANSACTIONS ON NETWORKING. He has served as a guest editor for the IEEE JSAC special issue on WDM network architectures, the *Computer Networks Journal* special issue on Broadband Internet Access, the *Journal of Communications and Networks* special issue on Wireless Ad-Hoc Networks, and the IEEE JOURNAL OF LIGHTWAVE TECHNOLOGY special issue on Optical Networks. He was the Technical Program co-chair for Wiopt 2006, IEEE INFOCOM 2007, and ACM MobiHoc 2007.

Chih-Ping Li received the B.S. degree from National Taiwan University, Taipei, Taiwan, R.O.C., in 2001, and the M.S. degree from the University of Southern California (USC), Los Angeles, in 2005, both in electrical engineering. He is currently working toward the Ph.D. degree at USC.

His research interests are in the areas of queueing theory and Markov decision processes.