A Unified Approach to Optimizing Performance in Networks serving Heterogeneous Flows

Ruogu Li, Lei Ying, Atilla Eryilmaz and Ness B. Shroff

***Abstract*— In this work, we study the control of communication networks in the presence of both inelastic and elastic traffic flows. The characteristics of these two types of traffic differ significantly. Hence, earlier approaches that focus on homogeneous scenarios with a single traffic type are not directly applicable. We formulate a new network optimization problem that incorporates the performance requirements of inelastic and elastic traffic flows. The solution of this problem provides us with a new queueing architecture, and distributed load balancing and congestion control algorithm with provably optimal performance. In particular, we show that our algorithm achieves the dual goal of maximizing the aggregate utility gained by the elastic flows while satisfying the demands of inelastic flows. Our base optimal algorithm is extended to provide better delay performance for both types of traffic with minimal degradation in throughput. It is also extended to the practically relevant case of dynamic arrivals and departures. Our solution allows for a controlled interaction between the performance of inelastic and elastic traffic flows. This performance can be tuned to achieve the appropriate design tradeoff. The network performance is studied both theoretically and through extensive simulations.**

I. Introduction

Over the last several years, we have witnessed the development of increasingly sophisticated optimization and control techniques to address a variety of resource allocation problems for communication networks (e.g. [2], [10], [18], [1], [9], [14], [21], [6], [19], [24], [12], [7], see [16], [8] for an overview). Much of this investigation has focused primarily on controllable or elastic traffic. However, networks are seeing a major growth in real-time traffic (voice and video), which is expected to consume an increasing fraction of the network services. This "inelastic" traffic does not lend itself to feedback control because of real-time constraints and its rate cannot be modulated without sacrificing quality. Thus, it is imperative that one develop efficient resource allocation strategies to jointly manage both inelastic and elastic traffic. Integration of elastic and inelastic flows in single-hop wireless systems has been studied [22], [3], [20]. In this paper, we consider general network topologies and develop a joint congestion control and load balancing mechanism that is fully distributed, and achieves high throughput and good delay characteristics.

In our model, inelastic traffic is given strict priority over elastic traffic since elastic traffic is usually more delay tolerant than its inelastic counterpart. Our goal is to balance the load of the inelastic traffic in the network such that the elastic traffic intelligently exploits the time varying residual capacity (the link capacity minus the capacity needed to serve the inelastic flows) at each link in the network. To see the potential gains of such an interaction, consider the network shown in Figure 1, which serves one inelastic and one elastic flow over links of capacity 20. Assume that the inelastic flow has a fixed rate of 20 and has two routes to divide its traffic over as shown in the figure. It can be seen that the rate distribution decision of the inelastic flow will significantly affect the elastic flow performance. If it divides its traffic equally amongst the two routes as in Figure 1, the elastic flow cannot achieve a rate more than 10. However, if the inelastic flow can steer more of its traffic over the uncongested route, more resources become available to the elastic traffic and it can achieve rates close to 20 as shown in Figure 2. With this intuition, we want to design a dynamic algorithm that automatically adapts the operation of inelastic and elastic flows to get the optimal performance. This requires a solution that seamlessly and distributively balances the load of the inelastic traffic across the network as well as injects enough elastic traffic into the network so that no capacity is wasted while preventing network overloading.

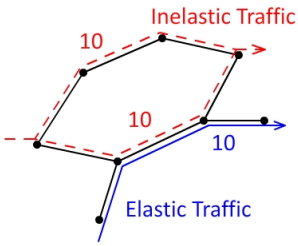

Fig. 1. Fixed Inelastic Traffic

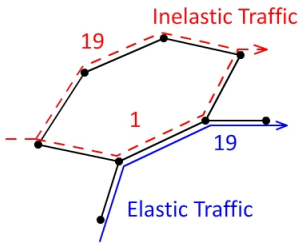

Fig. 2. Controllable Inelastic Traffic

We begin first by providing the system model and general assumptions. A priority queue for each link rather than each node is used in our queueing architecture. We then formulate a problem that attempts to maximize the utility of the elastic flows in the network subject to the constraint that the data requirements of the inelastic flows are met. We solve this problem via a two-step approach. First, we solve a simple version of the problem when the inelastic flow rates are deterministic. We then use the insights gained from that framework and extend the solution to the more general stochastic case. We then extend the work in two practically important directions. The first is to develop a virtual queue based solution that allows us to achieve low delays with a nominal and

Ruogu Li, Atilla Eryilmaz, and Ness B. Shroff are with the Department of Electrical and Computer Engineering at The Ohio State University, Columbus, Ohio USA, and Lei Ying is with the Department of Electrical and Computer Engineering at Iowa State University, Ames, Iowa USA.

Emails: {lir, eryilmaz, shroff}@ece.osu.edu, and leiying@engineering.iastate.edu.

controllable sacrifice in the throughput of the elastic flows. The second is to extend the solution in the presence of flow arrivals and departures, where certain elastic flows may be very short and may leave the system before the algorithm has the opportunity to converge.

We also present extensive simulations to demonstrate the interaction between the two types of flows under our proposed algorithms. In particular, we show that due to the dynamic nature of the load balancing mechanism implemented by inelastic sources, the elastic flows are able to push inelastic traffic onto less loaded routes and achieve higher rates. We show that this interaction maximizes the sum of the utilities of the elastic flows while satisfying the demands of the inelastic flows. We also compare the delay performance of our algorithm with and without the virtual queue implementation and illustrate that the virtual queue scheme can reduce the end-to-end delays significantly.

II. System Model and Objectives

We consider a fixed network represented by a graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{L}$ is the set of directed links. We assume that the capacity of link $l \in \mathcal{L}$ is c_l, and define the vector of link capacities as $\mathbf{c} := (c_l)_{l\in\mathcal{L}}$. Time is slotted in our system and the external packets arrive at the beginning of each time slot.

We consider the scenario where the network resources are shared by a set of *inelastic* and *elastic flows*, where a flow is defined by its source node and destination node. While the inelastic flow represents streaming traffic with fixed rates and stringent delay constraints such as real-time voice and video traffic, the elastic flow represents delay-tolerant traffic with adaptive rates such as non-real-time file sharing and email applications. The set of all flows in the network is denoted by $\mathcal{F}$, which is partitioned into two subsets, $\mathcal{F}_e$ and $\mathcal{F}_i$, where $\mathcal{F}_e$ is the set of elastic flows and $\mathcal{F}_i$ is the set of inelastic flows. Next, we describe the characteristics of inelastic and elastic flows in more detail.

Inelastic Flow: We let f_i denote an inelastic flow in the network with source s_i and destination d_i. Each inelastic flow f_i is associated with a fixed set of routes $\mathcal{R}_i$. The r^{th} route of this set is described by a vector $\mathbf{R}_i^{(r)}$ such that $\mathbf{R}_i^{(r)}[l] = 1$ if link $l \in \mathcal{L}$ is on that route, and zero otherwise. Let $x_i^{(r)}[t]$ be the number of injected packets on the r^{th} route of flow f_i at time slot t, and let $\mathbf{x}_i[t] := (x_i^{(r)}[t])_{f_i\in\mathcal{F}_i}^{\mathbf{R}^{(r)}\in\mathcal{R}_i}$ be the vector of inelastic flow packets injected on each route in slot t. Note that we slightly abuse our notation by using $\mathbf{x}_i$ to denote rate vector of all inelastic flows, while $x_i^{(r)}$ stands for the rate of flow $f_i \in \mathcal{F}_i$ over route $r \in \mathcal{R}_i$. We assume that the packet arrivals of the inelastic flow f_i follow a stochastic process $A_i[t]$ that is identically and independently distributed (*i.i.d.*) over time with a fixed mean rate, denoted by $a_i := \mathbb{E}(A_i[t])$, and a finite second moment, i.e. $\mathbb{E}(A_i^2[t]) < \infty$.

To clarify the difference between $A_i[t]$ and $(x_i^{(r)}[t])_{r\in\mathcal{R}_i}$, we note that $A_i[t]$ denotes the number of packets *generated* by flow f_i while $(x_i^{(r)}[t])_{r\in\mathcal{R}_i}$ describes the number of packets *injected* into the network to traverse each of the available routes of flow f_i. Thus, $A_i[t]$ is an uncontrollable stochastic process describing exogenous arrivals, whereas $(x_i^{(r)}[t])_{r\in\mathcal{R}_i}$ is controllable by the network algorithm.

For notational convenience, we define

$$z_l(\mathbf{x}_i[t]) := \sum_{f_i\in\mathcal{F}_i}\sum_{r=1}^{|\mathcal{R}_i|} x_i^{(r)}[t]\mathbf{R}_i^{(r)}[l]$$

to denote the total number of inelastic packets on link l for a given $\mathbf{x}_i[t]$.

Elastic Flow: We let f_e denote an elastic flow in the network with source s_e and destination d_e. We assume that each elastic flow f_e is associated with a single route $\mathbf{R}_e$, and we let $x_e[t]$ be the the number of injected packets of flow f_e in slot t. Similar to the inelastic case, we also define $\mathbf{x}_e[t] := (x_e[t])_{f_e\in\mathcal{F}_e}$ to be the vector of elastic flow rates in slot t, and

$$y_l(\mathbf{x}_e[t]) := \sum_{f_e\in\mathcal{F}_e} x_e[t]\mathbf{R}_e[l]$$

to denote the total number of elastic packets on link l. Associated with each elastic flow f_e there exists a utility function $U_e(\cdot)$ that measures the "satisfaction" of that flow as a function of its mean injection rate

$$\bar{x}_e := \lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1} x_e[t].$$

In the text, we use $\mathbf{x}[t] := (\mathbf{x}_i[t], \mathbf{x}_e[t])$ to denote the vector of inelastic and elastic packets injected into the network in slot t. Next, we provide a set of assumptions to be used later in the analysis:

Assumption 1: The elastic routing matrix $[\mathbf{R}_e]_{f_e\in\mathcal{F}_e}$ has full row rank, which guarantees that given $\mathbf{q}$, there exists a unique $\mathbf{p}$ such that $\mathbf{q} = ([\mathbf{R}_e]_{f_e\in\mathcal{F}_e})^T\mathbf{p}$.

Assumption 2: The inelastic arrival process $\{A_i[t]\}_{f_i\in\mathcal{F}_i}$ is such that there exists a vector $\mathbf{x}_i$ satisfying

$$\sum_{r=1}^{|\mathcal{R}_i|} x_i^{(r)} = a_i, \forall f_i \in \mathcal{F}_i, \qquad \text{and} \qquad z_l(\mathbf{x}_i) < c_l, \forall l \in \mathcal{L}.$$

This condition implies that the inelastic flows are supportable by the network, i.e., there exists a rate division of the inelastic flow rates over their available routes which can support the arriving traffic.

Assumption 3: The utility functions $\{U_e(x_e)\}_{f_e}$ are strictly concave, twice differentiable, and increasing functions. Such an assumption is commonly used to capture the diminishing returns to the elastic flows of an increase in the service rate.

Assumption 4: For each elastic flow $f_e \in \mathcal{F}_e$, its utility function $U_e(x)$ satisfies: for each $m > 0$ and $M \in [m, \infty)$, there exists $\tilde{c}_1, \tilde{C}_1, \tilde{c}_2$, and $\tilde{C}_2$, with

$$0 < \tilde{c}_1 < \tilde{C}_1 < +\infty, \qquad 0 < \tilde{c}_2 < \tilde{C}_2 < +\infty,$$

satisfying

$$\tilde{c}_1 \le U_e''(x) \le \tilde{C}_1, \qquad \tilde{c}_2 \le \left(U_e'^{-1}\right)'(x) \le \tilde{C}_2,$$

for all $x \in [m, M]$.

We note that Assumptions 3 and 4 on the utility functions are not restrictive and hold for the following class of utility functions $U(x) = w\, x^{(1-a)}/(1-a)$, for $a > 0$, which is known to characterize a large class of fairness concepts such as max-min fairness and weighted-proportional fairness ([23] and the references therein).

In subsequent discussions, when the distinction between real and non-real-time routes is unnecessary, we will simply refer to a route as $\mathbf{R}$ without any subscripts. Furthermore, for simplicity, we will use $z_l[t]$ for $z_l(\mathbf{x}_i[t])$ and $y_l[t]$ for $y_l(\mathbf{x}_e[t])$.

Queueing Architecture and Evolution: In our system, for each link $(i,j) \in \mathcal{L}$, a single priority queue is maintained at the transmitting node i, which holds all the packets whose routes traverse (i,j). Since the inelastic flows are expected to have more stringent delay constraints, their packets are always stored ahead of those of the elastic flows. We let $p_l[t]$ denote the queue length of the buffer associated with link l at the beginning of slot t, and define

$$q_{\mathbf{R}}[t] = \sum_{l \in \mathcal{L}} \mathbf{R}[l] p_l[t]$$

to be the total queue length on route $\mathbf{R}$. Notice that $p_l[t]$ and $q_{\mathbf{R}}[t]$ counts both the inelastic and elastic flows' packets.

During each time slot, the queue p_l evolves as

$$p_l[t+1] \quad = \quad (p_l[t] + y_l[t] + z_l[t] - c_l)^+, \tag{1}$$

where $x^+ = \max(0, x)$. This evolution is based on a *link-centric* decomposition ([16]) and implicitly assumes that packets injected into the source nodes by the flows, denoted by $\mathbf{x}[t]$, arrive at the downstream nodes instantaneously. In reality, packets will reach downstream nodes only after a queueing and propagation delay incurred in the intermediate nodes. It is shown in prior works ([25], [5], [26], [16]) that the inclusion of these dynamics do not affect the long-term stability and fairness characteristics of the system, and can be added to our queueing architecture by introducing a *regulator* queue before the queues associated with each link. Thus, in this work we use the evolution in (1) which possess a more tractable and cleaner form.

Definition 1 (Stability): We say that a queue $q_{\mathbf{R}}$ is *stable* if

$$\limsup_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left(q_{\mathbf{R}}[t]\right) \leq B, \tag{2}$$

where B is some finite positive value. We say that the *network is stable* if all aggregate queues $\{q_{\mathbf{R}}\}$ for both inelastic and elastic flows are stable.

Given the above network and traffic model, we aim to:

• Develop a mechanism that maximizes the total utility achieved by elastic flows, while satisfying the rate demands of the inelastic traffic. To that end, we design a joint congestion control and load balancing algorithm in Section III.

• Investigate means of extending our mechanism to improve the delay performance of both types of flows. To that end, we extend our joint algorithm in Section IV-A by adding appropriately constructed virtual queues with controllable parameters into the framework to achieve delay improvements.

• Analyze the performance of the joint algorithm under dynamic arrivals and departures to understand the effect of short-lived flows. To that end, we show in Section IV-B that such connection-level dynamics can be accommodated by our joint algorithm.

Before addressing these goals, we note that the load balancing component of our joint algorithm will dynamically control the distribution the inelastic flow rates over its available routes. Thus, the effect of inelastic traffic on the elastic traffic cannot be simply modeled as a constant decrease in the capacity of the network, and a more sophisticated approach is needed. In particular, the inelastic flow rates on each route must be balanced optimally to allow for the maximum utilization of the network resources by the competing elastic flows. We develop such an algorithm in the next section.

III. Joint Congestion Control and Load Balancing

In this section, we address our first main objective, i.e., that of developing an algorithm that provides maximum utilization of elastic traffic while guaranteeing the support of inelastic traffic. We start by describing our objective mathematically in the form of a stochastic optimization problem.

Stochastic Network Optimization (SNO) Problem:

$$\begin{aligned} \max_{\{\mathbf{x}[t] \geq 0\}_{t \geq 0}} \quad & \sum_{f_e \in \mathcal{F}_e} U_e(\bar{x}_e) \\ s.t. \quad & \text{The Queue Evolution as in}(1), \\ & \text{Network Stability as in}(2), \\ & \sum_{r=1}^{|\mathcal{R}_i|} x_i^{(r)}[t] = A_i[t], \forall f_i \in \mathcal{F}_i, \forall t > 0. \end{aligned} \tag{3}$$

We solve this problem by first analyzing a simpler deterministic fluid model in Section III-A. The solution to this fluid model will help in exposition as well as in providing insights on the solution of the above more complex problem. Then, we return in Section III-B to the stochastic problem.

A. Heuristic Fluid Model

In the fluid model scenario, all the dynamics and randomness are ignored, and the stochastic constraints are replaced with static constraints. In particular, the inelastic flow f_i is assumed to have a fixed arrival rate a_i, and the network stability condition is replaced by a condition on total link rate being no more than capacity. Then, the SNO problem reduces to the following problem in this scenario.

Fluid Network Optimization (FNO) Problem:

$$\begin{aligned} \max_{\mathbf{x} \geq 0} \quad & \sum_{f_e \in \mathcal{F}_e} U_e(x_e) \\ s.t. \quad & y_l(\mathbf{x}_e) + z_l(\mathbf{x}_i) \leq c_l, \forall l \in \mathcal{L} \qquad (4) \\ & \sum_{r=1}^{|\mathcal{R}_i|} x_i^{(r)} = a_i, \forall f_i \in \mathcal{F}_i \qquad (5) \end{aligned}$$

In our discussion, we will abbreviate the aggregate elastic and inelastic rates, $y_l(\mathbf{x}_e)$ and $z_l(\mathbf{x}_i)$, with y_l and z_l for brevity. We note that Condition (4) aims to capture the network stability condition in the fluid model by guaranteeing that the total load on a link is below the link capacity, and Condition (5) guarantees that inelastic flows receive enough bandwidth to satisfy its rate demands. Thus, the optimization problem is to maximize the sum of utilities of elastic flows when guaranteeing that inelastic flows are supported.

It is not difficult to show that the optimum value of FNO is an upper bound for the optimum value of SNO. To see this, note that any solution $\{\mathbf{x}[t]\}_{t\geq 0}$ that solve SNO must also satisfy $\bar{y}_l + \bar{z}_l \leq c_l$, where $\bar{y}_l := \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} y_l(\mathbf{x}[t])$, and $\bar{z}_l$ is defined similarly. Otherwise, the queue l cannot be stable. This is equivalent to condition (4) in FNO. Thus, FNO contains all the feasible points of SNO. In Section III-B, we will design an algorithm under which SNO can get arbitrarily close to the FNO solution, and thus guarantees the optimality of SNO.

We start by showing that there exists a unique $\mathbf{x}_e = \{x_e\}_{f_e\in\mathcal{F}_e}$, that solves the FNO problem under Assumption 2 and 3 [1].

Proposition 1: If Assumption 2 and 3 hold, then the $\mathbf{x}_e^* = \{x_e\}_{f_e\in\mathcal{F}_e}$ that solves the network optimization problem is unique.

Proof: The optimization problem has a unique solution because the utility functions are strictly concave, and constraints (4) and (5) are linear. ∎

To solve the FNO problem, we construct a partial Lagrangian. Define α_l to be the Lagrange multipliers associated with constraint (4). Then, the partial Lagrangian can be written as

$$\begin{aligned} L(\mathbf{x}_i, \mathbf{x}_e, \alpha) &= \sum_{f_e\in\mathcal{F}_e} U_e(x_e) - \sum_{l\in\mathcal{L}} \alpha_l (z_l + y_l - c_l) \\ &= \sum_{f_e\in\mathcal{F}_e} \left(U_e(x_e) - \left(\sum_{l:\mathbf{R}_e^{(r)}[l]=1} \alpha_l \right) x_e \right) \\ &\quad + \sum_{l\in\mathcal{L}} \alpha_l (c_l - z_l). \end{aligned}$$

Since FNO problem satisfies Slater's condition ([4]) due to Assumption 2, the strong duality holds. We can then conclude that there exists $\mathbf{x}^* := (\mathbf{x}_e^*, \mathbf{x}_i^*)$, and $\boldsymbol{\alpha}^* := (\alpha_l)_l$ such that

- $\mathbf{x}^*$ solves the FNO problem;
- $\mathbf{x}^* \in \arg\max_{\mathbf{x}\geq 0} L(\mathbf{x}_i, \mathbf{x}_e, \boldsymbol{\alpha}^*)$.

Note that

$$\begin{aligned} \max L(\mathbf{x}, \alpha^*) &= \max \sum_{f_e\in\mathcal{F}_e} \left(U_e(x_e) - \beta^*_{\mathbf{R}_e} x_e \right) \\ &\quad - \min \sum_{l\in\mathcal{L}} \alpha_l^* z_l + \sum_{l\in\mathcal{L}} \alpha_l^* c_l, \end{aligned}$$

where $\beta_{\mathbf{R}} := \sum_{l\in\mathcal{L}} \mathbf{R}[l]\alpha_l$. This decomposition suggests that

(i) The elastic flow f_e should allocate its rates such that

$$x_e^* = U_e'^{-1}\left(\beta^*_{\mathbf{R}_e^{(i)}}\right); \tag{6}$$

(ii) The inelastic flow f_i, should distribute its packets over its available routes $\{x_i^{*(r)}\}_{r\in\mathcal{R}_i}$ such that

$$\begin{aligned} \min \quad & \sum_{r=1}^{|\mathcal{R}_i|} x_i^{(r)} \beta^*_{\mathbf{R}_i^{(i)}} \\ s.t. \quad & \sum x_i^{(r)} = a_i. \end{aligned} \tag{7}$$

Since the optimization problem (7) has a linear objective, the following lemma holds ([4]):

Lemma 1: For any $\mathbf{R}_i^{(r)} \in \mathcal{R}_i$, we have:

- $\beta^*_{\mathbf{R}_i^{(r)}} = \beta^*_{\mathbf{R}_i^{(r')}}$ if $x_i^{*(r)} > 0$ and $x_i^{*(r')} > 0$; and
- $\beta^*_{\mathbf{R}_i^{(r)}} < \beta^*_{\mathbf{R}_i^{(r')}}$ if $x_i^{*(r)} > 0$ and $x_i^{*(r')} = 0$.

This lemma implies that considering an inelastic flow f_i, all routes in the optimal solution with a positive flow have the same value of β.

We note that α_l of FNO is closely associated with the queue length p_l of SNO, and correspondingly $\beta_{\mathbf{R}}$ of FNO is closely associated with the aggregate queue length on a route $q_{\mathbf{R}}$ of SNO. Such connections are revealed and exploited in several earlier works for designing different network algorithms (e.g. [14], [15], [6], [7], [24]). The following algorithm is a continuous-time version of the Lagrangian method for finding the optimum solution of FNO. This algorithm will later be used to solve the SNO. To distinguish the continuous-time evolution from the discrete-time evolution, we use (t) to denote continuous time index, while $[t]$ denotes discrete time index.

Joint Congestion Control and Load Balancing Algorithm for the FNO problem:

- Queue evolution for link l:

$$\dot{p}_l(t) := \frac{\mathrm{d}p(t)}{\mathrm{d}t} = (z_l(t) + y_l(t) - c_l)^+_{p_l(t)},$$

  where $(v(t))^+_{p(t)}$ is zero if $v(t) < 0$ and $p(t) = 0$; and $v(t)$ otherwise.
- Congestion Controller for elastic flow f_e:

$$x_e(t) = U_e'^{-1}\left(q_{\mathbf{R}_e}(t)\right).$$

- Load Balancing implemented for inelastic flow f_i:

$$\dot{x}_i^{(r)}(t) = \left(\bar{q}_i(t) - q_{\mathbf{R}_i^{(r)}}(t)\right)^+_{x_i^{(r)}(t)}, \tag{8}$$

  where $\bar{q}_i(t)$ satisfies

$$\sum_{r=1}^{|\mathcal{R}_i|} \left(\bar{q}_i(t) - q_{\mathbf{R}_i^{(r)}}(t)\right)^+_{x_i^{(r)}(t)} = 0, \tag{9}$$

  and $\sum_{r=1}^{|\mathcal{R}_i|} x_i^{(r)}(0) = a_i$.

---

[1] We note that the strict concavity assumption in Assumption 3 can be relaxed and our results can be extended to state that the elastic rates converges to the *set* of optimal rates rather than the *unique* optimum rate.

Remark: Note that the congestion control algorithm is motivated by equality (6). The load balancing algorithm (8) is motivated by Lemma 1. In particular, for each inelastic flow f_i, when the system reaches the equilibrium, we have $\dot{x}_i^{(r)}(t) = 0$ for all r. This implies that $q_{\mathbf{R}_i^{(r)}}(t) = \bar{q}_i(t)$ for $x_i^{(r)}(t) > 0$ and $q_{\mathbf{R}_i^{(r)}}(t) \geq \bar{q}_i$ for $x_i^{(r)}(t) = 0$. Thus, at the equilibrium point, $q_{\mathbf{R}_i^{(r)}}(t)$ satisfies Lemma 1. Furthermore, from (9), it is easy to see that

$$\sum_{r=1}^{|\mathcal{R}_i|} x_i^{(r)}(t) = a_i \qquad \text{for all } t. \tag{10}$$

The intuition behind the load balancing algorithm described above is to shift the inelastic flows to less heavily loaded routes to allow for the maximum network utilization for elastic flows.

Next, we will show the stability and optimality of our joint congestion control and load balancing algorithm.

Proposition 2: Under Assumption 1, 2 and 3, the joint congestion control and load balancing algorithm is globally asymptotically stable, i.e. $\lim_{t\to\infty} \mathbf{x}_e(t) = \mathbf{x}_e^*$ starting from any $\mathbf{x}(0)$, where $\mathbf{x}_e^*$ is the optimal solution to the FNO problem. Furthermore, (10) holds.

Proof: See [13] for the proof. ■

B. Stochastic Model

We now return to the original SNO problem with a minor variation:

SNO Problem with Parameter K (SNO-K):

$$\begin{aligned} \max_{\{\mathbf{x}[t]\geq 0\}_{t\geq 0}} \quad & \sum_{f_e\in\mathcal{F}_e} KU_e(\bar{x}_e) \\ s.t. \quad & \text{The Queue Evolution as in}(1), \\ & \text{Network Stability as in}(2), \\ & \sum_{r=1}^{|\mathcal{R}_i|} x_i^{(r)}[t] = A_i[t], \forall f_i \in \mathcal{F}_i, \forall t > 0, \end{aligned}$$

where K is a positive design parameter. We will see that K parameter is critical in eliminating the effect of randomness in the stochastic system on the long-term performance. Note that the solution to the SNO-K problem is independent of the value of K, and its optimum solution is identical to the solution of the SNO problem.

Motivated by the analysis in the fluid model, we propose the following joint congestion control and load balancing algorithm:

Joint Congestion Control and Load Balancing Algorithm for the SNO-K problem:

- Queue evolution for a link l:

$$p_l[t+1] = \left(p_l[t] + z_l[t] + y_l[t] - c_l\right)^+.$$

- Congestion Controller for elastic flow f_e:

$$x_e[t] = \min\left\{M, U_e'^{-1}\left(\frac{1}{K} q_{\mathbf{R}_e}[t]\right)\right\},$$

where M is a positive constant satisfies $M > 2\max_{l\in\mathcal{L}}\{c_l\}$.

- Load Balancing implemented for inelastic flow f_i:

$$\triangle x_i^{(r)}[t] = \left(\bar{q}_i[t] - q_{\mathbf{R}_i^{(r)}}[t]\right)^+_{x_i^{(r)}[t+1]},$$

or equivalently,

$$x_i^{(r)}[t+1] = \left(x_i^{(r)}[t] + \bar{q}_i[t] - q_{\mathbf{R}_i^{(r)}}[t]\right)^+,$$

where $\bar{q}_i[t]$ satisfies

$$\sum_{r=1}^{|\mathcal{R}_i|} \left(\bar{q}_i[t] - q_{\mathbf{R}_i^{(r)}}[t]\right)^+_{x_i^{(r)}[t+1]} = A_i[t+1] - A_i[t],$$

and $\sum_{r=1}^{|\mathcal{R}_i|} x_i^{(r)}[0] = A_i[0]$.

Remark: The factor $1/K$ in the congestion control equation comes from the factor K in the optimization problem. It can be interpreted as the aggressiveness factor of the elastic flow, as the congestion controller is inclined to inject more packets into the network with larger K. Also note that the load balancing implementation is slightly different from the fluid model version to accommodate the randomness in the arrival processes for inelastic flows. The update is modified to ensure that $\sum_{r=1}^{|\mathcal{R}_i|} x_i^{(r)}[t] = A_i[t]$ holds for all t.

The next proposition establishes the stability and optimality of the joint algorithm for the stochastic system.

Proposition 3: Under Assumptions 1, 2, 3 and 4, the joint congestion control and load balancing algorithm stabilizes the system in the sense that the Markov chain $(\mathbf{p}[t], \mathbf{x}_i[t])$ is positive recurrent with

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left(\left\|\mathbf{q}_{\mathbf{R}_e}[t] - \boldsymbol{\beta}^*_{\mathbf{R}_e}\right\|\right) \leq \frac{B + \epsilon\sigma K}{\epsilon},$$

and guarantees that the rate allocation satisfies,

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left(\left\|\mathbf{x}_e[t] - \mathbf{x}_e^*\right\|^2\right) \leq \frac{B}{\tilde{c}_3 K}.$$

Here $\mathbf{x}_e^*$ is the optimal solution to the SNO-K problem, σ is an arbitrarily chosen positive constant, and ϵ and $\tilde{c}_3$ are positive values.

Proof: See [13] for the proof. ■

Note that as the design parameter K increases, the rate converges to the optimal allocation at the cost of increased equilibrium queue-length levels. While such tradeoffs between optimality and delay is observed in earlier works under a single type of traffic (e.g. [19], [7]), in this work a new interaction is observed between inelastic and elastic traffic through the parameter K. In particular, larger values of K result in more aggressive elastic flows, resulting in larger queue-lengths on the links they traverse. This forces the inelastic flows to redistribute their flows to less loaded routes. This increases the utilization of the network, while causing more delay to inelastic flows. In order to provide better delay performance

to both types of traffic, in the next section we extend our base algorithm by using virtual queues.

IV. Extension of the Algorithm

In this section, we extend our joint congestion control and load balancing algorithm in two important directions: we first provide a virtual queue based solution that reduces the overall queue length with a negligible sacrifice in capacity. We then provide a solution that takes flow arrivals and departures into account.

A. Virtual Queue Algorithm

Inelastic applications are delay sensitive, hence we assume that packets from inelastic flows have strict priority over their elastic counterparts. Thus, the inelastic flows do not see the elastic flows in the queues they traverse. But in some cases a link might be critically loaded by the inelastic traffic itself, thus resulting in large delays. Also, elastic traffic may have some delay constraints that are non-negligible.

An effective way of reducing the experienced delay is by including virtual queues that are served at a fraction of the actual service rate, and by using the virtual queue-length values as prices ([11]). To that end, we introduce two types of virtual queues with parameters, ρ_1 and ρ_2, which control the total load and the inelastic flow load, respectively.

Here for simplicity, we go back to the fluid model to design and analyze the joint congestion control and load balancing algorithm using virtual queues. We would like to have $\tilde{\mathbf{x}}^* := (\tilde{\mathbf{x}}_e^*, \tilde{\mathbf{x}}_i^*)$ solve the following optimization problem.

FNO Problem with Virtual Queues (FNO-VQ):

$$\begin{aligned} \max_{\mathbf{x}\geq 0} \quad & \sum_{f_e\in\mathcal{F}_e} U_e(x_e) \qquad (11)\\ s.t. \quad & y_l(\mathbf{x}_e) + z_l(\mathbf{x}_i) \leq \rho_1 c_l, \forall l \in \mathcal{L} \\ & z_l(\mathbf{x}_i) \leq \rho_2 c_l, \forall l \in \mathcal{L} \\ & \sum_{r=1}^{|\mathcal{R}_i|} x_i^{(r)} = a_i, \end{aligned}$$

where $0 < \rho_2 \leq \rho_1 < 1$.

To guarantee the feasibility of the optimization problem above, we replace our earlier Assumption 2 with:

Assumption 5: There exists a $\mathbf{x}_i$ such that

$$\sum_{r=1}^{|\mathcal{R}_i|} x_i^{(r)} = a_i, \forall f_i \in \mathcal{F}_i, \qquad \text{and} \qquad z_l(\mathbf{x}_i) < \rho_2 c_l, \forall l \in \mathcal{L}.$$

To solve the FNO-VQ problem, we first introduce virtual queues for elastic and inelastic flows on each link respectively. The virtual queue length $\theta_l(t)$ for elastic flows evolutes as follows:

$$\dot{\theta}_l(t) = (z_l(t) + y_l(t) - \rho_1 c_l)^+_{\theta_l(t)}$$

The virtual queue length for inelastic flows $\gamma_l(t)$ evolves as follows:

$$\dot{\gamma}_l(t) = (z_l(t) - \rho_2 c_l)^+_{\gamma_l(t)}\,.$$

Note that when the total instantaneous traffic load is larger than $\rho_1 c_l$ or the inelastic traffic load is larger than $\rho_2 c_l$, the virtual queues will build up, and the network controller will reduce the traffic load.

Based on this virtual-queue scheme, we have the following joint congestion control and load balancing algorithm:

Joint Congestion Control and Load Balancing Algorithm for FNO-VQ problem:

- Virtual queue evolution for a link l:

$$\text{Elastic flows:} \quad \dot{\theta}_l(t) = (z_l(t) + y_l(t) - \rho_1 c_l)^+_{\theta_l(t)}\,;$$
$$\text{Inelastic flows:} \quad \dot{\gamma}_l(t) = (z_l(t) - \rho_2 c_l)^+_{\gamma_l(t)}\,.$$

- Congestion Controller for elastic flow f_e:

$$x_e(t) = U_e'^{-1}\left(s_{\mathbf{R}_e}(t)\right),$$

where $s_{\mathbf{R}_e}(t) = \sum_{l:\mathbf{R}_e[l]=1} \theta_l(t)$ is the aggregated virtual queue length of the elastic flow.

- Load Balancing implemented for inelastic flow f_i:

$$\dot{x}_i^{(r)}(t) = \left(\bar{\mu}_i(t) - \mu_{\mathbf{R}_i^{(r)}}(t)\right)^+_{x_i^{(r)}(t)},$$

where $\mu_{\mathbf{R}}(t) = \sum_{l:\mathbf{R}[l]=1}(\theta_l(t) + \gamma_l(t))$, $\bar{\mu}_i(t)$ satisfies

$$\sum_{r=1}^{|\mathcal{R}_i|} \left(\bar{\mu}_i(t) - \mu_{\mathbf{R}_i^{(r)}}(t)\right)^+_{x_i^{(r)}(t)} = 0$$

and $\sum_{r=1}^{|\mathcal{R}_i|} x_i^{(r)}(0) = a_i$.

Remark: In the above algorithm, note that the congestion control algorithm only responds to the virtual queues for elastic flows, but the load balancing algorithm responds to both the virtual queues for elastic flow and inelastic flows. Further, the actual queue length is not used in the algorithm.

In the following proposition, we show the equilibrium point of the algorithm with virtual queue is the optimal solution of (11).

Proposition 4: Under Assumptions 1, 3 and 5, the joint congestion control and load balancing algorithm for the FNO-VQ problem is globally asymptotically stable, i.e., $\lim_{t\to\infty} \mathbf{x}_e(t) = \tilde{\mathbf{x}}_e^*$ where $\tilde{\mathbf{x}}_e^*$ is the solution of the network optimization problem (11).

Proof: See [13] for the proof. ∎

The extension of this result to the stochastic scenario is omitted since it follows the same line of reasoning as in the joint congestion control and load balancing algorithm of Section III.

B. Dynamical File Arrivals and Departures

So far, we assumed long lasting flows. On the Internet, however, many elastic flows including email, web flows, etc are "mice" flows, i.e., short-lived flows. Thus, these short-lived flows might leave the network before the network reaches its

equilibrium point (the optimal rate allocation). In this section, we consider the scenario of dynamic flow generation and termination in networks with heterogenous traffic (see [17] and references therein). Note that since the rate allocation depends on the set of elastic flows, there exists no fixed rate allocation in this case. Thus, we focus on the dynamics of file arrivals and departures, where we assume that

(i) The number of inelastic flows are fixed.

(ii) Files belonging to elastic flows dynamically arrive and depart. We assume that files arrive according to independent Poisson processes with rate λ_e files/sec, the file sizes are independently and exponentially distributed, with mean file size $1/\mu_e$ bits. Further, let $n_e(t)$ denote the number of files belonging to flow f_e in the network at time t.

The following result states that our joint congestion control and load-balancing algorithm maximizes the network throughput region.

Proposition 5: Under Assumptions 1-3, the network is stable under the joint congestion control and load balancing algorithm if there exists $\hat{\mathbf{x}}_i$ such that $\sum_{r=1}^{|\mathcal{R}_i|} \hat{x}_i^{(r)} = a_i$, and for any $l \in \mathcal{L}$,

$$c_l - \sum_{f_i \in \mathcal{F}_i} \sum_{r=1}^{|\mathcal{R}_i|} \hat{x}_i^{(r)} \mathbf{R}_i^{(r)}[l] > \sum_{f_e \in \mathcal{F}_e} \frac{\lambda_e}{\mu_e} \mathbf{R}_e[l]. \tag{12}$$

Under Assumptions 1, 4, and 5, the network is stable under the joint congestion control and load balancing algorithm with virtual queues if there exists $\hat{\mathbf{x}}_i$ such that $\sum_{r=1}^{|\mathcal{R}_i|} \hat{x}_i^{(r)} = a_i$, and for any $l \in \mathcal{L}$, we have

$$\sum_{f_i \in \mathcal{F}_i} \sum_{r=1}^{|\mathcal{R}_i|} \hat{x}_i^{(r)} \mathbf{R}_i^{(r)}[l] \leq \rho_2 c_l,$$

$$\rho_1 c_l - \sum_{f_i \in \mathcal{F}_i} \sum_{r=1}^{|\mathcal{R}_i|} \hat{x}_i^{(r)} \mathbf{R}_i^{(r)}[l] > \sum_{f_e \in \mathcal{F}_e} \frac{\lambda_e}{\mu_e} \mathbf{R}_e[l]. \tag{13}$$

Proof: See [13] for the proof. ■

V. Simulation Results

In this section, we provide the simulation results for our algorithms under the stochastic model where the arrival process of the inelastic flow f_i is such that $A_i[t]$ is Poisson distributed with mean a_i for each t.

A. The Effect of the Aggressiveness of the Inelastic Flow

We noted in Section III-B that the factor K represents the 'aggressiveness' of the elastic flows. Also, it is revealed in Proposition 3 that K can be used to control the proximity to the optimal allocation. Here, we test these results for the case of proportionally fair allocation, which corresponds to having the utility function is chosen as ([23]): $U_e(x) = \alpha \ln x$, and thus $U_e'^{-1}(\frac{q}{K}) = \frac{\alpha K}{q}$.

In this first set of simulations, we considered the network shown in Figure 3 with the indicated link capacities and inelastic and elastic flows. Note that the arrival rate of the inelastic flow is $a_i = 15$ to be distributed over the two dashed routes.

The joint algorithm for the SNO-K problem is implemented for this network and the mean elastic rate allocation is computed for different values of K. Figure 4 illustrates the effect

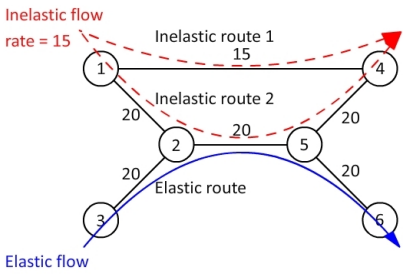

Fig. 3. Topology of the network

of K on the rates of the elastic flow and the distribution of the inelastic flow's rate over its available routes. We see that as the elastic flow becomes more aggressive, it achieves a higher throughput and thus consumes greater resource on the bottleneck link $(2, 5)$. As a reaction to the increased contention from the elastic flow, the load balancing mechanism of the inelastic flows automatically pushes more and more traffic of the inelastic flow onto route 1. When the bottleneck link is nearly fully utilized, the increase in aggressiveness results in small increase in the utilization of the link.

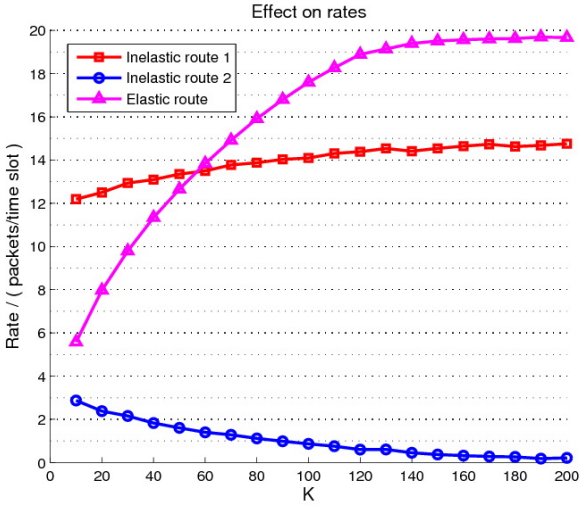

Fig. 4. Effect of K on the average rates on each route of Figure 3.

Note that although increasing the aggressiveness of the elastic flow will increase the utilization of the network, it will result in more delays on the network flows as the queue length over the whole network grows, as shown by Proposition 3. Proposition 3 also suggests that larger K resulting better

convergence to the optimal operating point, which is confirmed in the above simulation.

B. More Complex Topology

To illustrate other facets of our algorithm, we conducted our simulation in a more complicated network with different flow assignments. The topology of the network is shown in Figure 5. The capacity of all the links in the network is 20, and we used very aggressive elastic flows which have identical utility functions with $K = 200$ in our simulation, expecting close to optimal utilization (as shown in Figure 4).

We simulate a sequence of scenarios discussed in five phases. In Phase 1, two disjoint inelastic flows with the routes as shown in Figure 5 share the network, having rates $a_{i1} = 20$ and $a_{i2} = 10$. The average rates provided on each route by our joint algorithm are given in Figure 5.

When the two inelastic flows share a common bottleneck link in Phase 2, the load balancing algorithm will shift part of the traffic from the bottleneck link to yield the average rates given in Figure 6.

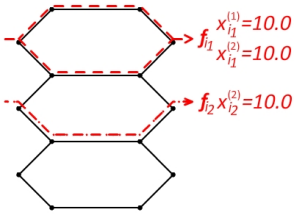

Fig. 5. Phase 1: Two inelastic flows with disjoint routes.

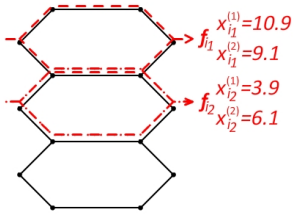

Fig. 6. Phase 2: Two inelastic flows with intersecting routes.

In Phase 3, an elastic flow enters the system and shares a link with f_{i2} as in Figure 7. We can see from the average rates given in the figure that this elastic flow not only has an effect on f_{i2} but also shifts the rate of f_{i1}. Here, it can be seen that the interactions between the flows becomes complex even for small networks, and it is not clear what the best allocation is. Yet, through our joint algorithm, f_{e1} is able to fully utilize all the resources available to it dynamically.

After adding another elastic flow f_{e2} into the network which is disjoint with all other flows in Phase 4 shown in Figure 8, we can see that it has no effect on the rates of all other routes, and it fully utilizes that route.

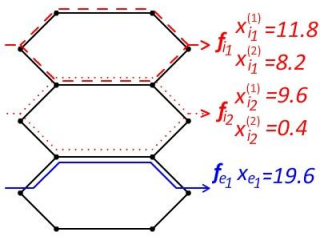

Fig. 7. Phase 3: Two intersecting inelastic flows, and one elastic flow that interacts with them.

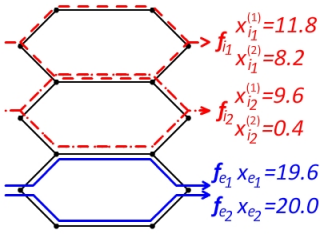

Fig. 8. Phase 4: Two intersecting inelastic flows, and two elastic flows with disjoint routes.

In Phase 5, a third elastic flow f_{e3} enters and shares common links with both f_{i1} and f_{i2}, as shown in Figure 9. We can see that since f_{e1} also shares links with f_{i2}, f_{e3} also has effect on it. It can be easily verified that $x_{e1} = x_{e3} = 15$ is the optimal operating point, and the average rates achieved by our algorithm is very close to optimal as predicted by Proposition 3.

To study the importance of dynamic load balancing, we also simulated a static rate distribution algorithm as a basis for comparison. This algorithm equally splits the inelastic traffic onto each of its routes (assume it is feasible in the network), and does the congestion control of the elastic flows in the same manner as in our algorithm. This algorithm is implemented for the scenario in Phase 5 with the average rates indicated in Figure 10. We see that due to the absence of dynamic load balancing, the elastic flows cannot utilize the network fully since the rates assigned to the inelastic flows are fixed. Under the logarithm utility function, this approach achieves a utility of $1.61\alpha K$ while our algorithm achieves $2.71\alpha K$ on the elastic flow f_{e3}.

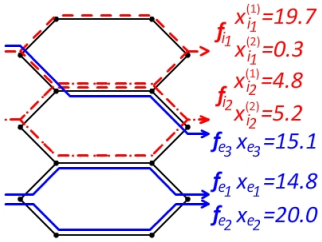

Fig. 9. Phase 5: A third elastic flow enters that intersects with two inelastic flows.

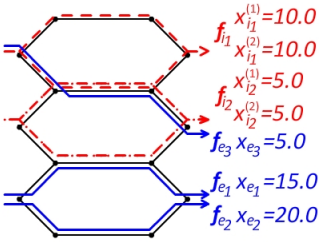

Fig. 10. Performance under static rate distribution for inelastic traffic.

C. Simulation using the Virtual Queue Algorithm

In this simulation, we use the joint congestion control and load balancing algorithm with virtual queue to show the impact of the virtual queue implementation on delay. The simulation is conducted in the network showing in Figure 11. The parameter ρ_1 was set to 0.95 and ρ_2 was set to 0.9 over all links.

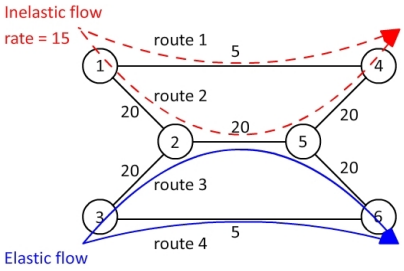

Fig. 11. Network topology for simulating the virtual queue algorithm

Table I compares the performance of the algorithms with and without virtual queues. As we can see from the table, under the original algorithm, Route 1 is critically loaded with inelastic flow traffic, resulting in a large delay. With the

TABLE I

RATE AND DELAY ON EACH ROUTES

	w/o virtual queue		w/ virtual queue	
	rate	delay	rate	delay
route 1	4.9955	40.320	4.4984	0.5405
route 2	9.7414	0.0440	10.502	0.0005
route 3	10.259	15.854	8.5118	1.0137
route 4	4.9950	77.461	4.7817	1.9765

implementation of the virtual queue, we manage to decrease the rate on Route 1, thus dropping the delay significantly. As one can observe from the table, the delay is greatly reduced for both elastic and inelastic traffic without a significant degradation in the rate of the elastic traffic. Thus, especially under critical loaded scenarios, virtual queue implementation can be used to get significant delay improvements.

VI. CONCLUSION

In this work, we consider the optimal control of networks that serve heterogeneous traffic types with diverse demands, namely inelastic and elastic traffic. We formulated a new network optimization problem and proposed a novel queueing architecture, and develop a distributed load balancing and congestion control algorithm with provably optimal performance. We also provided an important improvement to our joint algorithm to achieve better delay performance by introducing new design parameters (ρ_1, ρ_2) together with a set of virtual queues. We have also extended our algorithm to the case of dynamic arrivals and departures of the flows. Such a scenario is relevant to real-world operation as the real-world applications randomly initiate flows that lasts for a random duration.

Future research of this topic includes: (i) In this paper, we assume the link capacities are constant and there is no interference among links. One future direction is to extend our results to multi-hop wireless networks with fading channels and interference, and develop joint load-balancing/congestion control/routing/scheduling algorithms. (ii) In this paper, we considered a time-slotted system, and assumed that the network is perfectly synchronized. In real implementations, the algorithms will be executed in an asynchronous fashion. The impact of the asynchronism on the performance of the algorithms needs to be studied. (iii) We adopted a link-centric formulation, where packets are assumed to instantaneously arrive at all the links on their routes, and a source needs aggregated queue-lengths of its routes. An alternative is to consider a node-centric formulation, where packets are sequentially transferred, and a source only requires the information of the queues at the source.

REFERENCES

[1] E. Altman, T. Başar, and R. Srikant. Congestion control as a stochastic control problem with action delays. *Automatica*, pages 1937–1950, 1999. Special Issue on Control Methods for communication networks, V. Anantharam and J. Walrand, editors.

[2] J. Bolot and A. Shankar. Dynamic behavior of rate-based flow cntrol mechanisms. *ACM Comput. Commun. Rev.*, 20(2), 1992.

[3] S. Borst and N. Hegde. Integration of streaming and elastic traffic in wireless networks. In *Proc. INFOCOM*, May 2007.

[4] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, 2004.

[5] L. Bui, A. Eryilmaz, R. Srikant, and X. Wu. Joint asynchronous congestion control and distributed scheduling for wireless networks. Proceedings of IEEE Infocom 2006.

[6] A. Eryilmaz and R. Srikant. Fair resource allocation in wireless networks using queue-length based scheduling and congestion control. In *Proceedings of IEEE Infocom*, volume 3, pages 1794–1803, Miami, FL, March 2005.

[7] A. Eryilmaz and R. Srikant. Resource allocation of multi-hop wireless networks. In *Proceedings of International Zurich Seminar on Communications*, February 2006.

[8] L. Georgiadis, M. J. Neely, and L. Tassiulas. *Resource Allocation and Cross-Layer Control in Wireless Networks*. 2006. Foundations and Trends in Networking.

[9] K. Kar, S. Sarkar, and L. Tassiulas. A simple rate control algorithm for maximizing total user utillity. In *Proceedings of IEEE INFOCOM*, Anchorage, Alaska, 2001.

[10] F. P. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: Shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49:237–252, 1998.

[11] S. Kunniyur and R. Srikant. Analysis and design of an adaptive virtual queue algorithm for active queue management. *IEEE/ACM Transactions on Networking*, pages 286–299, April 2004. An earlier version appeared in Proc. ACM Sigcomm 2001.

[12] J. Lee, R. R. Mazumdar, and N. Shroff. Non-convex optimization and rate control for multi-class services in the internet. *IEEE/ACM Trans. on Networking*, 13(4):827 – 840, Aug 2005.

[13] R. Li, L. Ying, A. Eryilmaz, and N. B. Shroff. A unified approach to optimizing performance in networks serving heterogeneous flows, 2008. Technical Report, available online at http://www.ece.osu.edu/∼eryilmaz/ElasticInelasticReport.pdf.

[14] X. Lin and N. Shroff. Joint rate control and scheduling in multihop wireless networks. In *Proceedings of IEEE Conference on Decision and Control*, Paradise Island, Bahamas, December 2004.

[15] X. Lin and N. Shroff. The impact of imperfect scheduling on cross-layer rate control in multihop wireless networks. In *Proceedings of IEEE Infocom*, Miami, FL, March 2005.

[16] X. Lin, N. B. Shroff, and R. Srikant. A tutorial on cross-layer optimization in wireless networks. *IEEE Journal on Selected Areas in Communications, special issue on Nonlinear Optimization of Communication Systems*, 14(8), Aug 2006.

[17] X. Lin, N. B. Shroff, and R. Srikant. On the connection-level stability of congestion-controlled communication networks. *IEEE Transactions on Information Theory*, 2008. To appear.

[18] S. H. Low and D. E. Lapsley. Optimization flow control, I: Basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 7:861–875, December 1999.

[19] M. Neely, E. Modiano, and C. Li. Fairness and optimal stochastic control for heterogeneous networks. In *Proceedings of IEEE Infocom*, pages 1723–1734, Miami, FL, March 2005.

[20] S. Patil and G. Veciana. Managing resources and quality of service in heterogeneous wireless systems exploiting opportunism. *IEEE/ACM Trans. Netw.*, 15(5):1046–1058, 2007.

[21] D. Qiu and N. B. Shroff. A predictive flow control scheme for efficient network utilization and QoS. *IEEE/ACM Transactions on Networking*, 12(1), February 2004.

[22] S. Shakkottai and A. Stolyar. Scheduling algorithms for a mixture of real-time and non-real-time data in hdr. In *in Proceedings of 17th International Teletraffic Congress (ITC-17*, pages 793–804, 2001.

[23] R. Srikant. *The Mathematics of Internet Congestion Control*. Birkhäuser, Boston, MA, 2004.

[24] A. Stolyar. Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm. *Queueing Systems*, 50(4):401–457, 2005.

[25] X. Wu and R. Srikant. Regulated maximal matching: A distributed scheduling algorithm for multi-hop wireless networks with node-exclusive spectrum sharing. In *Proceedings of* IEEE Conference on Decision and Control., 2005.

[26] L. Ying, R. Srikant, A. Eryilmaz, and G. E. Dullerud. Distributed fair resource allocation in cellular networks in the presence of heterogeneous delays. In *Proceedings of International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT)*, 2005.