

A Integrated Approach to Optimizing Presentation in Networks Portion Various Flows

¹Sandhya Kondapureddy, ²Rajeshwar Singh Bondili

^{1,2}Dept. of CSE, Swarna Bharathi Institute of Science and Technology, Khammam, AP, India

Abstract

In this work, we develop the mathematical framework and novel design methodologies needed to support such heterogeneous requirements and propose provably optimal network algorithms that account for the multilevel interactions between the flows. To that end, we first formulate a network optimization problem that incorporates the above throughput and service prioritization requirements of the two traffic types. We note that the coexistence of such diverse flows creates complex interactions at multiple levels (e.g., flow and packet levels), which prevent the use of earlier design approaches that dominantly assume homogeneous traffic. We study the optimal control of communication networks in the presence of heterogeneous traffic requirements. Specifically, we distinguish the flows into two crucial classes: inelastic for modeling high-priority, delay-sensitive, and fixed-throughput applications; and elastic for modeling low-priority, delay-tolerant, and throughput-greedy applications. We then develop a distributed joint load-balancing and congestion control algorithm that achieves the dual goal of maximizing the aggregate utility gained by the elastic flows while satisfying the fixed throughput and prioritization requirements of the inelastic flows. Next, we extend our joint algorithm in two ways to further improve its performance: in delay through a virtual queue implementation with minimal throughput degradation and in utilization by allowing for dynamic multipath routing for elastic flows. A unique characteristic of our proposed dynamic routing solution is the novel two-stage queueing architecture it introduces to satisfy the service prioritization requirement.

Keywords

Optimal Network Algorithms, Heterogeneous, Mathematical Framework

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 for an overview). Much of this investigation has focused primarily on optimizing functions of long-term performance metrics such as throughput subject to network stability. Two types of traffic can be distinguished: elastic traffic with controllable packet injection rates generated by file transfer or other delay-tolerant applications, and inelastic traffic with fixed packet injection rates generated by delay-sensitive applications. In [27], the coexistence of inelastic and elastic flows has also been considered in a more general setup. However, previous utility maximization-based solutions do not distinguish inelastic packets and elastic packets at the packet level. Thus, the inelastic packets need to compete with elastic packets for link bandwidths, so these two types of flows have comparable delay performance. Yet, inelastic flows model delay-sensitive traffic and must be served with higher priority as they traverse the network. This prioritization decouples the inelastic packets and elastic packets at the link (or, equivalently, packet) level and will result in small delays for inelastic flows. Note that even though two types of flows are decoupled at the link

level, to provide high utilization for elastic traffic, the inelastic flows must smartly distribute their load among their available routes. To that end, we developed our algorithm to maximize the network utility defined by elastic flows under the prioritization, which provides new coupling methods at the flow level that are different from previous utility maximization-based solutions. We believe the main contributions of this work to be the following.

- The mathematical formulation of the utility maximization problem for elastic rate control subject to inelastic traffic requirements of fixed rate and service prioritization.
- The development of a distributed joint load-balancing and rate-control algorithm that gives strict service priority to inelastic packets while guaranteeing optimal resource utilization for elastic traffic. The description and the optimality of this algorithm are provided for both the fluid model and the actual stochastic network.
- The extension of the base algorithm to a virtual queue-based operation that enables further delay reduction for both traffic types with a nominal and controllable sacrifice in the network utilization.
- The relaxation of the static route assumption for the elastic flows to achieve higher utilization of the network resources through dynamic, multipath routing while maintaining the prioritization requirements. This leads to a novel two-stage queueing architecture that complies with the prioritization requirements of the design.

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 is c_l , and define the vector of link capacities as $(c_l)_{l \in \mathcal{L}}$. 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 traffic with fixed rates and stringent delay constraints such as voice and video streaming, the elastic flow represents delay-tolerant traffic with adaptive rates such as non-real-time file sharing and e-mail 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 t th route of this set is described by a vector $\mathbf{R}_i^{(r)}$ such that if link $l \in \mathcal{L}$ is on that route, and zero otherwise. Let $x_i^{(r)}[t]$ be the flow rate of f_i on route r at time t .

number of injected packets on the t th route of flow at time slot t , $\mathbf{x}_i[t]$ be the vector of inelastic flow packets injected on each route in slot t . 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 λ_i and a finite second moment, i.e., $E[A_i[t]^2] < \infty$. To clarify the difference between $A_i[t]$ and $\left(x_i^{(r)}[t]\right)_{r \in \mathcal{R}_i}$, we note that $A_i[t]$ denotes the

number of packets generated by flow f_i while $\left(x_i^{(r)}[t]\right)_{r \in \mathcal{R}_i}$ 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 $\left(x_i^{(r)}[t]\right)_{r \in \mathcal{R}_i}$ is controllable by the network algorithm. For notational convenience, we define

$$\sum_{i \in \mathcal{I}} \sum_{f_i \in \mathcal{F}_i} \sum_{r=1}^{|\mathcal{R}_i|} x_i^{(r)}[t] \mathbf{R}_i^{(r)}[t]$$

$$z_l(\mathbf{x}_i[t])$$

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 s_e, t_e with source s_e and destination d_e . In Section IV-B, we will consider dynamic routing for the elastic flow, but for now we will concentrate on the fix route case to get some insight about the load balancing. Here, 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 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

$$\sum_{e \in \mathcal{E}} x_e[t] \mathbf{R}_e[l] y_l(\mathbf{x}_e[t])$$

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

We also assume that the sources of the elastic flows are always infinitely backlogged.

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 has full row rank, which guarantees that given there exists a unique

$$\mathbf{PQ} = ([\mathbf{R}_e]_{f_e \in \mathcal{F}_e})^T \mathbf{P} \text{ such that } \mathbf{PQ} = \mathbf{I}.$$

Assumption 2: The inelastic arrival process is such that there exists a vector λ_i satisfying

$$\sum_{i \in \mathcal{I}} \sum_{f_i \in \mathcal{F}_i} \sum_{r=1}^{|\mathcal{R}_i|} x_i^{(r)}[t] = a_i \forall f_i \in \mathcal{CN} \text{ } l \in \mathcal{L} \text{ } f_i \text{ and } z_l(\mathbf{x}_i)$$

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 that can support the arriving traffic.

Assumption 3: The utility functions 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.

In subsequent discussions, when the distinction between real and nonreal-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 $l \in \mathcal{L}$ each link (i, j) , a single-priority queue is maintained at the transmitting node, 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, giving inelastic traffic full priority over its elastic counterpart. We let denote the queue length of the buffer associated with link l at the beginning of slot t and define

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

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

During each time slot, the queue evolves as

$$q_{\mathbf{R}}[t+1] = (p_l[t] - c_l)_+ + p_l[t] + y_l[t] + z_l[t] \quad (1)$$

where $\max(0, x)$. This evolution is based on a link-centric decomposition ([18]) 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 [5, 18, 29-30] that the inclusion of these dynamics does not affect the long-term stability and fairness characteristics of the system. In particular, a regulator queue for each flow can be added to our queueing architecture before the queues associated with each link as the same architecture in [5]. The prioritization in the per-link queue does not affect the queue evolution, hence the results in [5] apply in our model with priority. Thus, in this work we use the evolution in (1), which possesses a more tractable and cleaner form.

Definition 1 (Stability): We say that a queue is stable if $\limsup_{T \rightarrow \infty} \frac{1}{T} E(q_{\mathbf{R}}[T]) \leq B$ 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 this network and traffic model, we aim to do the following.

- Develop a mechanism that maximizes the total utility achieved by elastic flows while giving strict priority to and 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.
- By relaxing the single-route constraint (and thus removing Assumption 1) on the elastic flow and adapting a new queueing architecture, we develop a joint congestion control, dynamic routing, and load-balancing algorithm in Section IV-B.

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. 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 Section III.

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:

$$\max_{\mathbf{x}} \sum_{f_e \in \mathcal{F}_e} U_e(\mathbf{x}_e) \quad \text{s.t.} \quad \text{Queue Evolution as in (1)}$$

$$\sum_{i \in \mathcal{R}_i} x_i^{(r)}[t] \quad \text{Network Stability as in (2)}$$

$$A_i[t] \forall f_i \in \mathcal{F}_i \geq 0$$

(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, 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:

Note that

$$\max_{\mathbf{x} \geq 0} \sum_{f_e \in \mathcal{F}_e} U_e(\mathbf{x}_e) \quad \text{s.t.} \quad c_l \forall l \in \mathcal{L} \quad \sum_{i \in \mathcal{R}_i} x_i^{(r)} = a_i \forall f_i \in \mathcal{F}_i \quad \max L(\mathbf{x}, \alpha^*) = \max$$

$$\sum_{f_e \in \mathcal{F}_e} (U_e(\mathbf{x}_e) - \beta_{\mathbf{R}}^* \mathbf{x}_e) y_l(\mathbf{x}_e) + z_l(\mathbf{x}_i) \leq \quad (4)$$

$$f_i. \quad (5)$$

In our discussion, we will abbreviate the aggregate elastic and inelastic rates, $y_l(\mathbf{x}_e)$ and $z_l(\mathbf{x}_i)$, with \bar{y}_l and \bar{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 that solves SNO must also satisfy $\bar{y}_l + \bar{z}_l \leq c_l$, where \bar{y}_l and \bar{z}_l is defined similarly. Otherwise, the queue 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 $\{\mathbf{x}_e^*\}_{f_e \in \mathcal{F}_e}$ that solves the FNO problem under Assumptions 2 and 3.2

Proposition 1: If Assumptions 2 and 3 hold, then the $\mathbf{x}_e^* = \{\mathbf{x}_e^*\}_{f_e \in \mathcal{F}_e}$ 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 to be the Lagrange multipliers associated with constraint (4). Then, the partial Lagrangian can be written as

$$\sum_{m \in \mathcal{N}} r_{mi}^d[t] - \sum_{j \in \mathcal{N}} r_{ij}^d[t] \quad \text{where } \beta_{\mathbf{R}}. \text{ This decomposition suggests the following conditions.}$$

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

$$\mathbf{x}_e = (\beta_{\mathbf{R}}^*)_{f_i} \left\{ x_i^{*(r)} \right\}_{r \in \mathcal{R}_i} U_e^{-1} \quad (6) \quad (ii)$$

The inelastic flows should distribute its packets over its

$$\sum_{i \in \mathcal{R}_i} x_i^{(r)} \beta_{\mathbf{R}_i}^{*(r)} \text{ available routes such that } \min$$

$$\text{s.t.} \quad a_i \sum_{i \in \mathcal{R}_i} x_i^{(r)}$$

(7) Since the optimization problem (7) has a linear objective, the

$\mathbf{R}_i^{(r)} \in \mathbf{R}_i^{*(r)}$ following lemma holds [4].

Lemma 1: For any \mathcal{R}_i , we have:

$$0 < x_i^{(r')} \beta_{\mathbf{R}_i}^{*(r')} x_i^{(r)} \beta_{\mathbf{R}_i}^{*(r)} \quad \text{if and;}$$

$0 < x_i^{(r)} \beta_{\mathbf{R}_i}^{*(r)} x_i^{(r')} \beta_{\mathbf{R}_i}^{*(r')}$ if and. 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 $\beta_{\mathbf{R}_i}$. We note that of FNO is closely associated with the queue length of SNO, and correspondingly $\beta_{\mathbf{R}}$ of FNO is closely associated with the aggregate queue length on a route of SNO. Such connections are revealed and exploited in several earlier works for designing different network algorithms. 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.

$$L(\mathbf{x}_i, \mathbf{x}_e, \alpha) = \sum_{f_e \in \mathcal{F}_e} U_e(\mathbf{x}_e) - \sum_{l \in \mathcal{L}} \alpha_l (z_l + y_l - c_l) \left(U_e(\mathbf{x}_e) - \sum_{i \in \mathcal{R}_i} x_i^{(r)}[t] = 1 \right)$$

Joint Congestion Control and Load-Balancing Algorithm for the FNO Problem:

$$\sum_{f_e \in \mathcal{F}_e} p_l(t) \dot{p}(t) = (z_l(t) + y_l(t) - c_l) + p_l(t) \quad \text{Queue evolution for link } l$$

$\sum_{l \in \mathcal{L}} \alpha_l c_l z_l$ Since the FNO problem satisfies Slater's condition [4] due to Assumption 2, the strong duality holds. We can then conclude that there exists $(\mathbf{x}_e^*, \mathbf{x}_i^*)$, and (α_l) such that:

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

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 converge to the set of optimal rates rather than the unique optimum rate.

$-\min_{l \in \mathcal{L}} \alpha_l^* z_l \sum_{l \in \mathcal{L}} \alpha_l^* c_l$
 where is zero if $v(t)$ and $p(t)$; and $v(t)$ otherwise.

$(v(t))_{p(t) < 0} = 0$. Congestion controller for elastic flow f_e

$= U_e^{-1}(q_{\mathbf{R}_e}(t)) x_e(t)$
 $f_i \dot{x}_i^{(r)}(t) = (\bar{q}_i(t) + q_{\mathbf{R}_i^{(r)}}(t)) x_i^{(r)}(t)$. Load balancing implemented for inelastic flow (8)

$\sum_{r=1}^{|R_i|} q_{\mathbf{R}_i^{(r)}}(t) x_i^{(r)}(t) \sum_{r=1}^{|R_i|} x_i^{(r)}(0) A_i(\bar{q}_i(t))$
 = and

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:

• Scheduling with Strict Prioritization:

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

$\dot{x}_i^{(r)}(t) x_i^{(r)}(t) = 0$ for all r . This implies that $q_{\mathbf{R}_i^{(r)}}(t) = \bar{q}_i(t)$ for

and for $x_i^{(r)}(t)$. Thus, at the equilibrium point, satisfies Lemma 1. Furthermore, from (9), it is easy to For each link $l \in \mathcal{L}$, we serve packets from, with strict priority to inelastic packets, which leads to the following queue evolution:

$$\dot{c_l p_l + 1} \leq (p_l[t] + z_l[t] + y_l[t] c_l)_+ p_l[t]$$

f_e Congestion controller for elastic flow

$$x_e[t] \leq \min \left\{ M U_e^{-1} \left(\sum_{r=1}^{|R_e|} q_{\mathbf{R}_e^{(r)}}(t) \right) \right\}$$

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

Proposition 2: Under Assumptions 1–3, the joint congestion

$\mathbf{p}(t), \mathbf{q}(t) \rightarrow (\mathbf{x}_e^*, \mathbf{x}_i^*)$ as $t \rightarrow \infty$ control and load-balancing algorithm is globally asymptotically stable, i.e.,

$(\mathbf{x}_e^*, \mathbf{x}_i^*, \alpha^*, \beta^*)$, where is an optimal prime-dual solution to the FNO problem. Furthermore, (10) holds.

Proof: The proof is provided in Appendix A.

B. Stochastic Model

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

SNO Problem with Parameter K (SNO-K):

$\sum_{f_e \in \mathcal{F}_e} K U_e(\bar{x}_e) (\mathbf{x}[t] \geq 0)_{t \geq 0}$
 s.t. Queue Evolution as in (1)

$\sum_{i \in \mathcal{I}} x_i^{(r)}[t]$ Network Stability as in (2)

$$A_i[t] \forall f_i \in \mathcal{F}_i \geq 0$$

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.

where is a positive constant satisfies

$$M M > 2 \max_{l \in \mathcal{L}} \{c_l\}.$$

• Load balancing implemented for inelastic flow f_i

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

or equivalently

$$+ 1 \leq (x_i^{(r)}[t] + \bar{q}_i[t] q_{\mathbf{R}_i^{(r)}}[t]) x_i^{(r)}[t]$$

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

$$\sum_{r=1}^{|R_i|} q_{\mathbf{R}_i^{(r)}}[t] x_i^{(r)}[t+1] \leq A_i[t+1] A_i[t] \sum_{r=1}^{|R_i|} x_i^{(r)}[0] A_i[0] \bar{q}_i[t]$$

Remark: The factor in the congestion control equation comes from the factor 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. 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 $\sum_{r=1}^{|R_i|} x_i^{(r)}[t] A_i[t]$ holds for all.

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

Proposition 3: Under Assumptions 1–4, the joint congestion control and load-balancing algorithm stabilizes the system in the sense that the Markov chain is positive recurrent with

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

and guarantees that the rate allocation satisfies

Proof: See [15] for the proof.

Note that as the design parameter increases, the rate converges to the optimal allocation at the cost of increased equilibrium queue-length levels. In this work a new interaction is observed between inelastic and elastic traffic through the parameter. In

particular, larger values of 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 allows the dynamic routing for the elastic flow.

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. However, 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 nonnegligible. To that end, we introduce two types of virtual queues with parameters to the previous optimization problem, and, 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 $(\tilde{x}_e^*, \tilde{x}_i^*)$ solve the following optimization problem.

FNO Problem With Virtual Queues (FNO-VQ):

$$\sum_{f_e \in \mathcal{F}_e} U_e(x_e) \max_{x \geq 0}$$

$$y_l(x_e) + z_l(x_i) \leq \rho_1 c_l \quad l \in \mathcal{L}, s.t.$$

$$\leq \rho_2 c_l \quad l \in \mathcal{L}, \sum_{i \in \mathcal{R}_l} x_i^{(r)} \leq z_l(x_i)$$

$$= a_i(11)$$

$$0 < \rho_2 < \rho_1 < 1 \text{ where } \leq.$$

To guarantee the feasibility of this optimization problem, we replace our earlier Assumption 2 with Assumption 5.

Assumption 5: There exists an ϵ such that \mathcal{F}_i and $z_l(x_i)$

This can be viewed as admission control is done for the inelastic

flow after we choose the parameter to ensure the feasibility of the problem.

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 evolves as follows:

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

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

$$\gamma_l(t) (z_l(t) - \rho_2 c_l) + \gamma_l(t) =$$

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

- Virtual queue evolution for a link l

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

- Congestion controller for elastic flow f_e

$$= U_e^{-1}(s_{R_e}(t)) x_e(t)$$

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

- Load balancing implemented for inelastic flow f_i

$$= \left(-\mu_i(t) - \mu_{R_i^{(r)}}(t) \right) x_i^{(r)}(t) \dot{x}_i^{(r)}(t)$$

$$= + \gamma_l(t) \mu_i(t) \text{ where } \mu_{R_i}(t) = \sum_{l: R_i[l]=1} (\theta_l(t) + \gamma_l(t))$$

satisfies

$$\sum_{i \in \mathcal{R}_l} \left(-\mu_i(t) - \mu_{R_i^{(r)}}(t) \right) x_i^{(r)}(t) \leq \sum_{r=1}^{|\mathcal{R}_l|} x_i^{(r)}(0) \quad \forall l$$

= and

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 flows and inelastic flows. Furthermore, 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 virtual

$(\tilde{x}_e^*, \tilde{x}_i^*, \tilde{\theta}^*, \gamma^*)$ system under the joint congestion control and load-balancing algorithm for the FNO-VQ problem is globally asymptotically stable, i.e. where is an optimal primal-dual solution of the network optimization problem (11).

Proof: See for the proof.

Remark: Proposition 4 implies the stability of the virtual system. Since the virtual system is run in a network with smaller link capacity, it is intuitively reasonable to expect the real queue lengths be stable. While this intuition is confirmed in numerical studies, we do not claim the stability of the real system under this algorithm.

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. Dynamic Routing for Elastic Flows

Here, we keep the assumptions of the inelastic flows and relax the single-route constraint on the elastic flows. Each elastic flow f_e now is associated with a source node \mathcal{N} and a destination node \mathcal{N} . The injection rate of the flow f_e is denoted by x_s^d , and $U_s^d(x_s^d)$ is the utility function.

In this model, we adapt a new queueing architecture. Each node maintains both real queues and virtual queues (counters), which are identical structural-wise. We focus on the virtual queues, which are neater to analyze.

Within the virtual queues, there are two levels of queues. The first level of the virtual queues are for each destination for elastic flows, denoted by $s_i^d[t]$, each node maintains such queues. Let $r_{ij}^d[t]$ denote the promised service rate to elastic flows that are destined to on link (i, j) , and let

$$\sum_{d \in \mathcal{N}} r_{ij}^d[t] y_l[t]$$

denotes the total promised service of elastic flows on link l . These counters evolve as

$$\left(s_i^d[t] \sum_{m \in \mathcal{N}} r_{mi}^d[t] - \sum_{j \in \mathcal{N}} r_{ij}^d[t] \right) + 1] + x_i^d[t] + s_i^d[t]$$

(12) The second level of virtual queues consists of a virtual queue $\in \mathcal{L}$ for each outgoing link, denoted by $p_{ij}[t]$ where (i, j) , which counts packets from both inelastic and elastic flows.

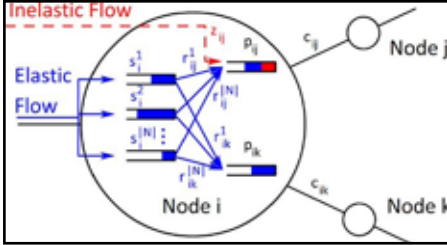


Fig. 1: Queueing Architecture for Dynamic Routing

Packets of the inelastic flows are always “stored” ahead of those of the elastic flows. These counters evolve as

$$p_i[t+1] = y_i[t] + z_i[t] \quad (13)$$

Remark 1: Note that in the virtual queues (counters), virtual packets “bypass” the queueing in the second level of virtual queues and directly enter the corresponding first-level virtual queues of the next node, while $r_{ij}^d[t]$ actual packets are forwarded within the node to the second level of queues, waiting to be transmitted onto the corresponding link to the downstream node. The decisions made by the algorithm are based on the virtual queue information, which is a “prediction” of the actual queue lengths. We will see that the actual queues evolve in the same way when the system reaches its equilibrium point.

Remark 2: The evolution of $p[t]$ given in (13) implicitly assumes that the inelastic packets injected into the source nodes arrive at the downstream nodes immediately. Prior works show that this assumption does not affect the long-term performance of the system. Since the inelastic traffic enters the second-level queues directly, the results of those works still apply to the inelastic flows. In particular, a regulator queue for each inelastic flow can be added to our queueing architecture before the second-level queues associated with each link following the same approach as in . The queueing architecture described is illustrated in Fig. 1. For simplicity, we consider the fluid model scenario for dynamic routing.

FNO Problem With Dynamic Routing (FNO-DR):

$$\max_{\mathbf{x} \geq 0} U_e(\mathbf{x}_e) \text{ s.t. } \forall i \in \mathcal{N} \sum_{r=1}^R x_i^{(r)} \sum_{f \in \mathcal{F}_e} y_i(\mathbf{r}) + z_i(\mathbf{x}_i) \leq a_i \quad (14)$$

$$= a_i \quad \forall f \in \mathcal{F}_i \quad (15)$$

$$+ \sum_{m \in \mathcal{N}} r_{mi}^d \leq \sum_{j \in \mathcal{N}} r_{ij}^d \quad \forall i, d \in \mathcal{N}, i \neq d \quad (16)$$

$$\leq p \in \mathcal{L}, d \in \mathcal{N} \quad (17) \quad r_{ij}^d \quad \forall (i, j) \quad (18)$$

where (17) is the flow conservation constraint, and (18) is a fixed bound on r_{ij}^d , where $\max_{i \in \mathcal{L}} \{c_i\}$.

Based on the new queueing architecture and the FNO-DR problem, using the Lagrange multiplier approach, we have the following joint congestion control, dynamic routing, and load-balancing algorithm.

Remark: In the algorithm above, the rate control of the elastic traffic only needs the virtual queue information of the source node, while the load balancing of the inelastic uses the virtual queue information on the whole route. The dynamic routing

algorithm for elastic traffic is similar to the back-pressure algorithm, while the backlog difference between the two nodes of a link is offset by, discouraging the elastic flows from using the links that are in the route of the inelastic flows. Also, we note that this back-pressure-like algorithm is applied to the elastic traffic only, and the strict priority of the inelastic traffic is still maintained.

In the following proposition, we show the equilibrium point of the algorithm for dynamic routing is the optimal solution of (14).

Proposition 5: Under Assumptions 2 and 3, the virtual $\mathbf{p}(t), \mathbf{s}(t) (\hat{\mathbf{x}}_e^*, \hat{\mathbf{x}}_i^*, \lim_{t \rightarrow \infty} (\mathbf{x}_e(t), \mathbf{x}_i(t), \mathbf{p}^*, \hat{\mathbf{s}}^*))$ system under the joint congestion control, dynamic routing, and load-balancing algorithm for the FNO-DR problem is globally asymptotically stable, i.e.,

$(\hat{\mathbf{x}}_e^*, \hat{\mathbf{x}}_i^*, \mathbf{p}^*, \hat{\mathbf{s}}^*)$ where is an optimal primal-dual solution of the network optimization problem (14).

Proof: The proof is provided in Appendix B.

Remark: Note that the virtual system will evolve to the optimal point according to the equations above, while the actual system may follow a slightly different trajectory. This is because during the transient period of evolution, the total incoming rate

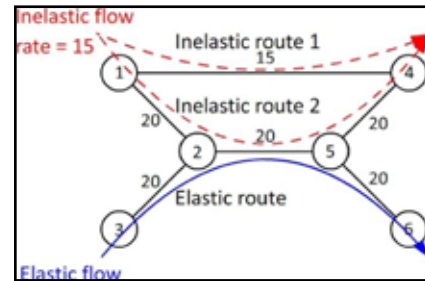


Fig. 2: Topology of the Network

of inelastic flow $\mathbf{z}(t)$ and the elastic flow $\mathbf{r}(t)$ into the second level of the queue (cf. Fig. 1) is allowed to be larger than the capacity of the link. Thus, when the system made routing decisions according to the virtual values (the estimated number of packets in the queues), there may not be enough packets to transmit in the actual queues. However, when the virtual system converges to its equilibrium point, the assigned service rate r_{ij}^{*d} will be feasible as shown in Proposition 5, and therefore the actual system will also be able to provide the same rate as the virtual one. Thus, at the equilibrium point, the stability of the virtual system implies the stability of the actual system.

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 for each .

A. Effect of the Aggressiveness of the Inelastic Flow

We noted in Section III-B that the factor represents the “aggressiveness” of the elastic flows. Also, it is revealed in Proposition 3 that 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 [26]: $U_e(\mathbf{x})$, and thus. In this first set of simulations, we considered the network shown in Fig. 2 with the indicated link capacities and inelastic and elastic flows. Note that the arrival rate of the inelastic flow is 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 I , illustrates the effect of I 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.

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.

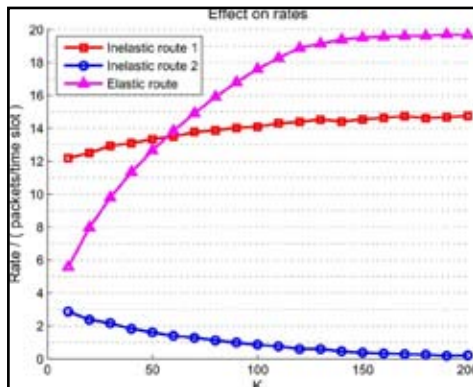


Fig. 3: Effect of I on the Average Rates on Each Route of fig. 2.

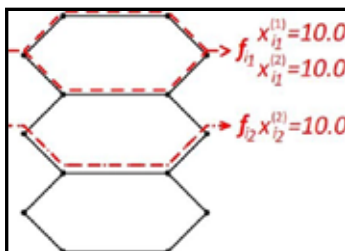


Fig. 4: Phase 1: Two Inelastic Flows With Disjoint Routes.

Proposition 3 also suggests that larger 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. $a_{i1}=20, a_{i2}=10$. We simulate a sequence of scenarios discussed in five phases. In Phase 1, two disjoint inelastic flows with the routes as shown in Fig. 4 share the network, having rates and. The average rates provided on each route by our joint algorithm are given in Fig. 4.

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 Fig. 5.

In Phase 3, an elastic flow enters the system and shares a link with f_{i2} as in Fig. 6. Here, it can be seen that the interaction 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 operate dynamically close to the full capacity of all the resources available to it.

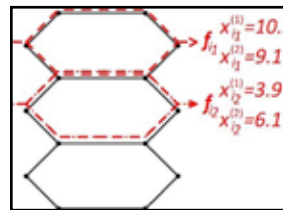


Fig. 5: Phase 2: Two Inelastic Flows With Intersecting Routes.

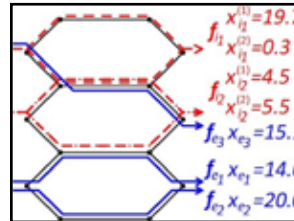


Fig. 6: Phase 5: A Third Elastic Flow Enters, Which Intersects with Two Inelastic Flows

After adding another elastic flow f_{e2} into the network, which is disjoint with all other flows in Phase 4 shown in Fig. 7, we can see that it has no effect on the rates of all other routes, and it fully utilizes that route.

In Phase 5, a third elastic flow f_{e3} enters and shares common links with both f_{i1} and f_{i2} , as shown in Fig. 8. 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 is the optimal operating point, and the average rates achieved by our algorithm are 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 Fig. 9. We see that due to the absence of dynamic load balancing, the elastic flows

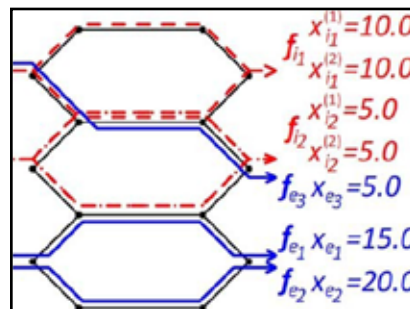


Fig. 9: Performance Under Static Rate Distribution for Inelastic Traffic

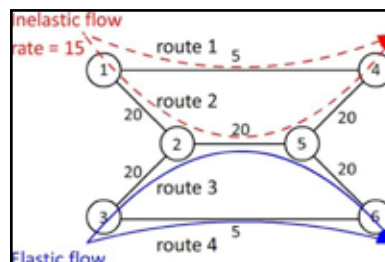


Fig. 10: Network Topology for Simulating the Virtual Queue Algorithm

1.61aK2.71aK 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, while our algorithm achieves on the elastic flow f_{e3} .

In this sequence of simulations, we see complex interactions between the elastic and inelastic flows and observed that our algorithm achieves optimal performance distributively. While we have so far focused on the average achieved rates, in Fig. 10, we also show the time evolution of the flow rates on each route as the network goes through the five phases. We see in this figure that the rates converge to their optimal value quickly, which may be important under dynamic conditions.

C. Simulation Using the Virtual Queue Algorithm

$\rho_1 \rho_2$ 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 Fig. 11. The parameter was set to 0.95, and was set to 0.9, over all links.

Table I compares the performance with and without virtual queues. First, we note that on link (2,3), though it is critically loaded by two types of flows, the inelastic flow has a much smaller delay compared to the elastic flow due to the strict prioritization. Also, as we can observe, under the original algorithm, route 1 is critically loaded by inelastic flow, resulting in a large delay. With the virtual queue implementation, 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 achieve significant delay improvements. Also, for the elastic flows, the delay drops significantly in the tradeoff of a small loss in utilization.

Table 1: Rate and Delay on Each Route

	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

VI. Conclusion

We formulated a new network optimization problem, proposed a novel queueing architecture, and developed a distributed load-balancing and congestion control algorithm with provably optimal performance. In this work, we consider the optimal control of networks that serve heterogeneous traffic types with diverse demands, namely inelastic and elastic traffic. 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 allowing elastic flows to choose their routes dynamically, which will further utilize the resource available in the network.

Future research of this topic includes the following. 1) One future direction is to extend our results to multihop wireless networks with fading channels and interference and develop joint load-balancing/congestion control/routing/scheduling algorithms. 2) Here, we considered a time-slotted system and assumed that the network is perfectly synchronized. The impact of possible asynchronism on the algorithm performance needs to be studied. 3) We adopted a link-centric formulation, which

assumes instantaneous arrivals of the packets at all the links on their 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. 4) So far, we have focused on the stability and long-term guarantees for the traffic types. We aim to investigate oscillatory behavior and delay characteristics in our future work. 5) In this work, we assume the routes and the supportability of the inelastic flow are given. Developing corresponding routing and admission control mechanism will make it complete.

APPENDIX A

PROOF OF PROPOSITION 2

$$V(\mathbf{p}(t), \mathbf{x}_i(t)) = \sum_{l \in \mathcal{L}} (p_l(t) - \alpha_l^*)^2 \quad \text{Define a Lyapunov function} \quad (19)$$

where the first inequality holds because \leq for any $l \in \mathcal{L}$, and α_l^* if $y_l^* \geq z_l^*$, if and only if $y_l^* + z_l^*$; the second equality holds due to Assumption 1.

Without loss of generality, we assume that for all

then we have

$$\begin{aligned} & (z_l(t) \\ & + \sum_{|\mathcal{R}_i|: f_i \in \mathcal{F}_i} \sum_{r=1}^R (x_i^{(r)}(t) - \alpha_i^*) \sum_{l \in \mathcal{L}} (p_l(t) - \alpha_l^*) p_l(t) + |\mathcal{R}_i|: f_i \in \mathcal{F}_i} \sum_{r=1}^R \\ & (x_i^{(r)}(t) - \alpha_i^*) \sum_{l \in \mathcal{L}} (p_l(t) - \alpha_l^*) (z_l(t) + y_l(t) - c_l + p_l(t) + l \in \mathcal{L} \\ & (p_l(t) - \alpha_l^*) - z_l^*) \leq \sum_{|\mathcal{R}_i|: f_i \in \mathcal{F}_i} \sum_{r=1}^R (x_i^{(r)}(t) - \alpha_i^*) \sum_{l \in \mathcal{L}} (p_l(t) - \alpha_l^*) \\ & \sum_{l \in \mathcal{L}} (p_l(t) - \alpha_l^*) (z_l(t) - z_l^*) = \sum_{|\mathcal{R}_i|: f_i \in \mathcal{F}_i} \sum_{r=1}^R (x_i^{(r)}(t) - \alpha_i^*) \sum_{l \in \mathcal{L}} (p_l(t) - \alpha_l^*) \\ & - q_{\mathbf{R}_i^{(r)}}(t) + \sum_{l \in \mathcal{L}} (p_l(t) - \alpha_l^*) \sum_{|\mathcal{R}_i|: f_i \in \mathcal{F}_i} \sum_{r=1}^R (x_i^{(r)}(t) - \alpha_i^*) \mathbf{R}_i^{(r)}[l] \\ & \text{where } \mathbf{x}_i^* \text{ and } \alpha_i^* \text{ satisfy conditions (i) and (ii). Note that } \mathbf{x}_i^* \text{ and } \alpha_i^* \text{ may not be unique. Then, we have} \\ & (\bar{q}_i(t) + q_{\mathbf{R}_i^{(r)}}(t))_{x_i^{(r)}(t)} \sum_{l \in \mathcal{L}} (p_l(t) - \alpha_l^*) (z_l(t) - z_l^*) + \sum_{l \in \mathcal{L}} (p_l(t) - \alpha_l^*) \\ & (y_l(t) + z_l^* - c_l) + |\mathcal{R}_i|: f_i \in \mathcal{F}_i} \sum_{r=1}^R (x_i^{(r)}(t) - \alpha_i^*) \sum_{|\mathcal{R}_i|: f_i \in \mathcal{F}_i} \sum_{r=1}^R (x_i^{(r)}(t) \\ & - \alpha_i^*) (q_{\mathbf{R}_i^{(r)}}(t) - \beta_{\mathbf{R}_i^{(r)}}^*) = \sum_{|\mathcal{R}_i|: f_i \in \mathcal{F}_i} \sum_{r=1}^R (x_i^{(r)}(t) - \alpha_i^*) \sum_{l \in \mathcal{L}} (p_l(t) - \alpha_l^*) \\ & - \beta_{\mathbf{R}_i^{(r)}}^*) \leq \sum_{(a): f_i \in \mathcal{F}_i} \left((\bar{q}_i(t) - \beta_{\mathbf{R}_i^{(1)}}^*) \sum_{|\mathcal{R}_i|: f_i \in \mathcal{F}_i} \sum_{r=1}^R (x_i^{(r)}(t) - \alpha_i^*) \right) \quad (21) \\ & + |\mathcal{R}_i|: f_i \in \mathcal{F}_i} \sum_{r=1}^R (x_i^{(r)}(t) - \alpha_i^*) \sum_{l \in \mathcal{L}} (p_l(t) - \alpha_l^*) (\bar{q}_i(t) + q_{\mathbf{R}_i^{(r)}}(t))_{x_i^{(r)}(t)} \sum_{l \in \mathcal{L}} \end{aligned}$$

$$\begin{aligned}
& p_l(t) - \alpha_l^* (z_l(t) + y_l(t) - c_l) + |\mathcal{R}_i| \sum_{f_i \in \mathcal{F}_i} \sum_{r=1}^R \left(x_i^{(r)}(t) - x_i^{*(r)} \right) \times \dots \\
& \sum_{|\mathcal{R}_i| \sum_{f_i \in \mathcal{F}_i} \sum_{r=1}^R \left(x_i^{(r)}(t) - x_i^{*(r)} \right) \left(\bar{q}_i(t) - q_{\mathbf{R}_i^{(r)}}(t) \right) + |\mathcal{R}_i| \sum_{f_i \in \mathcal{F}_i} \sum_{r=1}^R \sum_{l \in \mathcal{L}} \left(p_l(t) - \alpha_l^* \right) \mathbf{R}_i^{(r)} [l] \left(x_i^{(r)}(t) - x_i^{*(r)} \right) = \sum_{|\mathcal{R}_i| \sum_{f_i \in \mathcal{F}_i} \sum_{r=1}^R \left(x_i^{(r)}(t) - x_i^{*(r)} \right) \left(\bar{q}_i(t) - q_{\mathbf{R}_i^{(r)}}(t) \right) \left(\bar{q}_i(t) - q_{\mathbf{R}_i^{(r)}}(t) \right) x_i^{(r)}(t) = (b)
\end{aligned}$$

Next, we show the negativity of the drift term by term. First, we have

$$\begin{aligned}
& \sum_{l \in \mathcal{L}} (p_l(t) - \alpha_l^*) + z_l^* c_l (y_l(t) - y_l^*) \leq \\
& \sum_{l \in \mathcal{L}} (p_l(t) - \alpha_l^*) (y_l(t) - y_l^*) \leq \\
& (\mathbf{p}^T(t) - \alpha^{*T}) \mathbf{R}_e (\mathbf{x}_e(t) - \mathbf{x}_e^*) = \\
& (\mathbf{q}^T(t) - \beta^{*T}) (\mathbf{x}_e(t) - \mathbf{x}_e^*) = \\
& \sum_{f_e \in \mathcal{F}_e} (U'_e(x_e(t)) - U'_e(x_e^*)) = \\
& (x_e(t) - x_e^*) \times
\end{aligned}$$

$$0 \leq \quad (20)$$

where inequality (a) holds due to Lemma 1, and equality (b) holds due to the equality (10).

Thus, from inequalities (20) and (21), we can conclude that

$$\begin{aligned}
& 0 \leq \\
& \leq \\
& \text{for all } t. \text{ Furthermore, from inequality (20), we also know that } \\
& = 0, \mathbf{x}_i(t)) \quad \text{if and only if } \mathbf{x}_e(t) = \mathbf{x}_e^*.
\end{aligned}$$

APPENDIX B

PROOF OF PROPOSITION 5

We construct a partial Lagrangian to solve the FNO-DR problem. Let λ_l and μ_i^d be the Lagrange multipliers associated with the constraints (15) and (17), respectively. Furthermore,

$$\begin{aligned}
& \left(\mu_i^{*d} \left(\sum_m r_{mi}^{*d} - \sum_j r_{ij}^{*d} \right) \right) - \sum_{(i,j),d} r_{ij}^d p_{ij} - (32) \text{ term} - \sum_{i,d} \left(s_i^d - s_j^d \right) \\
& \text{to the rest of the terms, } \left(s_i^d - s_j^d \right) \text{ we get} \\
& = - \left(\sum_{(i,j),d} r_{ij}^d \left(\mu_i^{*d} - \mu_j^{*d} \right) \leq (a) - \sum_{(i,j),d} p_{ij} r_{ij}^{*d} + \left(\sum_{i,d} x_i^{*d} s_i^d - \sum_{(i,j),d} r_{ij}^d \left(s_i^d - s_j^d - p_{ij} \right) \right) \leq (b) \sum_{i,d} \sum_{m} \left(\mu_i^{*d} - \mu_j^{*d} \right) p_{ij} r_{ij}^{*d} - \sum_{(i,j),d} r_{ij}^d \left(s_i^d - s_j^d - p_{ij} \right) \sum_{i,d} r_{ij}^{*d} \left(s_i^d - s_j^d - p_{ij} \right) \\
& (33)
\end{aligned}$$

$$+ (35) = \sum_{(i,j),d} p_{ij} r_{ij}^d - \sum_{(i,j),d} p_{ij} r_{ij}^{*d} - \sum_{(i,j),d} \lambda_{ij}^* r_{ij}^d + \sum_{(i,j),d} \lambda_{ij}^* r_{ij}^{*d} \left(\sum_{i,d} r_{mi}^d - \sum_j r_{ij}^d \right)$$

$$\left(\sum_{i,d} r_{mi}^d - \sum_j r_{ij}^d \right) \quad (36) \quad (37)$$

$$\left(\sum_{i,d} x_i^{*d} s_i^d + s_{im}^d \left(x_i^{*d} s_i^d \right) \right) \quad (38)$$

$$\begin{aligned}
& - \sum_{i,d} \left(x_i^{*d} \left(\mu_i^{*d} + \mu_i^{*d} \left(\sum_m r_{mi}^{*d} - \sum_j r_{ij}^{*d} \right) \right) + \sum_{i,d} \left(\mu_i^{*d} \left(\sum_m r_{mi}^{*d} - \sum_j r_{ij}^{*d} \right) \right) \left(\mu_i^{*d} \left(\sum_m r_{mi}^{*d} - \sum_j r_{ij}^{*d} \right) \right) \right) \\
& \left(\sum_j r_{ij}^{*d} \right) \left(\mu_i^{*d} \left(\sum_m r_{mi}^{*d} - \sum_j r_{ij}^{*d} \right) \right)
\end{aligned} \quad (39)$$

$$\left(\sum_{i,d} r_{mi}^{*d} - \sum_j r_{ij}^{*d} \right) \quad (40)$$

(41) Note that (39) by (27). Combine the first term in (36) with (38), the third term with (41) and (37) with (40), and we

$$\begin{aligned}
& (33) + (35) = - \sum_{(i,j),d} p_{ij} r_{ij}^{*d} + \left(\sum_{i,d} x_i^{*d} s_i^d - \sum_{(i,j),d} r_{ij}^d \left(s_i^d - s_j^d \right) \left(\sum_{(i,j),d} r_{ij}^d \right) \right) \\
& r_{ij}^{*d} \text{ get}
\end{aligned}$$

where (a) holds since r_{ij}^{*d} maximizes over all, thus

$$\begin{aligned}
& \sum_{(i,j),d} r_{ij}^d \left(\mu_i^{*d} - \mu_j^{*d} - \lambda_{ij}^* \right) \\
& \left(\sum_{(i,j),d} r_{ij}^{*d} \left(\mu_i^{*d} - \mu_j^{*d} - \lambda_{ij}^* \right) r_{ij}^d \left(\mu_i^{*d} - \mu_j^{*d} - \lambda_{ij}^* \right) \right) \sum_{(i,j),d} r_{ij}^d \\
& \geq
\end{aligned}$$

and (b) holds follow from (23). Thus, we conclude that $0 \leq$ for all t , and the equality holds if and only if $\mathbf{x}_e(t) = \mathbf{x}_e^*$.

References

- [1] E. Altman, T. Basar, R. Srikant, "Congestion control as a stochastic control problem with action delays", Automatica, vol. 35, pp.1937–1950, 1999.
- [2] J. Bolot, A. Shankar, "Dynamic behavior of rate-based flow control mechanisms", ACM Comput. Commun. Rev., Vol. 20, No. 2, pp. 35–49, 1992.
- [3] S. Borst, N. Hegde, "Integration of streaming and elastic traffic in wireless networks", in Proc. IEEE INFOCOM, May 2007, pp. 1884–1892.
- [4] S. Boyd, L. Vandenberghe, "Convex Optimization", Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [5] L. Bui, A. Eryilmaz, R. Srikant, X. Wu, "Joint asynchronous congestion control and distributed scheduling for wireless networks", in Proc. IEEE INFOCOM, 2006.
- [6] A. Eryilmaz, R. Srikant, "Fair resource allocation in wireless networks using queue-length based scheduling and congestion control", in Proc. IEEE INFOCOM, Miami, FL, Mar. 2005, Vol. 3, pp. 1794–1803.
- [7] A. Eryilmaz, R. Srikant, "Resource allocation of multi-hop wireless networks", in Proc. Int. Zurich Seminar Commun., Feb. 2006, pp.

90–93.

- [8] A. Eryilmaz, R. Srikant, J. R. Perkins, "Stable scheduling policies for fading wireless channels", IEEE/ACM Trans. Networking, Vol. 13, No. 2, pp. 411–425, Apr. 2005.
- [9] L. Georgiadis, M. J. Neely, L. Tassiulas, "Resource Allocation and Cross-Layer Control in Wireless Networks", ser. Foundations and Trends in Networking. Hanover, MA: now, 2006.
- [10] P. Hande, S. Zhang, M. Chiang, "Distributed rate allocation for in- elastic flows", IEEE/ACM Trans. Netw., Vol. 15, No. 6, pp. 1240–1253, Dec. 2007.
- [11] K. Kar, S. Sarkar, L. Tassiulas, "A simple rate control algorithm for maximizing total user utility", in Proc. IEEE INFOCOM, Anchorage, AK, 2001, Vol. 1, pp. 133–141.
- [12] F. P. Kelly, A. Maulloo, D. Tan, "Rate control in communication networks: Shadow prices, proportional fairness and stability", J. Oper. Res. Soc., Vol. 49, pp. 237–252, 1998.
- [13] S. Kunniyur, R. Srikant, "An adaptive virtual queue (AVQ) algo- rithm for active queue management", IEEE/ACM Trans. Netw., Vol. 12, No. 2, pp. 286–299, Apr. 2004.
- [14] J. Lee, R. R. Mazumdar, N. Shroff, "Non-convex optimization and rate control for multi-class services in the internet," IEEE/ACM Trans. Netw., Vol. 13, No. 4, pp. 827–840, Aug. 2005.
- [15] R. Li, A. Eryilmaz, L. Ying, N. B. Shroff, "A unified approach to optimizing performance in networks serving heterogeneous flows", in Proc. IEEE INFOCOM, 2009, pp. 253–261.
- [16] X. Lin, N. Shroff, "Joint rate control and scheduling in multihop wireless networks", in Proc. IEEE CDC, Paradise Island, Bahamas, Dec. 2004, Vol. 2, pp. 1484–1489.
- [17] X. Lin, N. Shroff, "The impact of imperfect scheduling on cross- layer rate control in multihop wireless networks", in Proc. IEEE IN-FOCOM, Miami, FL, Mar. 2005, Vol. 3, pp. 1804–1814.
- [18] X. Lin, N. B. Shroff, R. Srikant, "A tutorial on cross-layer opti- mization in wireless networks," IEEE J. Sel. Areas Commun., Vol. 24, No. 8, pp. 1452–1463, Aug. 2006.
- [19] S. H. Low, D. E. Lapsley, "Optimization flow control, I: Basic al- gorithm and convergence", IEEE/ACM Trans. Netw., Vol. 7, No. 6, pp. 861–875, Dec. 1999.



Sandhya Kondapureddy pursuing M.Tech in the Department Of Computer Science and Technology. Her research areas include Computer Networks, Information Security, Data Mining and Data Warehousing.



Rajeshwar Singh Bondili working as an Assistant Professor in the Department Of Computer Science and Engineering. His research areas include Cloud Computing, Data Mining and Data Warehousing, Java.