

Value-aware Resource Allocation for Service Guarantees in Networks

Parimal Parag, Srinivas Shakkottai and Jean-François Chamberland

Department of Electrical and Computer Engineering

Texas A&M University, College Station, TX 77843-3128, USA

Email: {parimal, sshakkot, chmbrlnd}@tamu.edu

Abstract—The traditional formulation of the total value of information transfer is a multi-commodity flow problem. Here, each data source is seen as generating a commodity along a fixed route, and the objective is to maximize the total system throughput under some concept of fairness, subject to capacity constraints of the links used. This problem is well studied under the framework of network utility maximization and has led to several different distributed congestion control schemes. However, this idea of value does not capture the fact that flows might associate value, not just with throughput, but with link-quality metrics such as packet delay, jitter and so on. The traditional congestion control problem is redefined to include individual source preferences. It is assumed that degradation in link quality seen by a flow adds up on the links it traverses, and the total utility is maximized in such a way that the quality degradation seen by each source is bounded by a value that it declares. Decoupling source-dissatisfaction and link-degradation through an “effective capacity” variable, a distributed and provably optimal resource allocation algorithm is designed, to maximize system utility subject to these quality constraints. The applicability of our controller in different situations is illustrated, and results are supported through numerical examples.

I. INTRODUCTION

Recent years have seen an enormous growth in demand for Internet access, with applications ranging from personal use to commercial and military operations. Several of these applications are sensitive to a “quality” of packet delivery. For instance, although archiving data transfer can tolerate long delays, voice over Internet protocol (VoIP) is very sensitive to latency. Between these two extreme examples lies a spectrum of applications with varying service requirements, e.g. electronic commerce, video conferencing and online gaming. All these applications require the allocation of enough network resources for satisfactory performance.

The design of efficient network control systems demands that end-user value be taken into consideration when allocating resources. The Internet architecture is built around the concept of a *flow*, which is a transfer of data between a fixed source-destination pair. How do we quantify the value of such a flow? The classical formulation of the total value of information transfer is a multi-commodity flow problem, in which each data source is seen as generating a commodity along a fixed route, and the objective is to maximize the total throughput under some concept of fairness, subject to capacity constraints

of the links used [1]–[4]. If the flow from source r has a rate $x_r \geq 0$ and the system utility associated with such a flow is represented by a concave, increasing function $U_r(x_r)$, the objective is

$$\begin{aligned} \max \quad & \sum_{r \in \mathcal{S}} U_r(x_r) \\ \text{s.t.} \quad & y_l \leq c_l, \forall l \in \mathcal{L} \end{aligned} \quad (1)$$

where \mathcal{S} is the set of sources, \mathcal{L} the set of links, c_l the capacity of link $l \in \mathcal{L}$. Also let R be the routing matrix with $R_{lr} = 1$ if the route associated with source r uses link l . The load on link l is $y_l = \sum_{r \in \mathcal{S}} R_{lr} x_r$. Note that we refer to flows and sources interchangeably; if there are multiple flows between a source and a destination, we simply give them different names. This is a convex optimization problem that is well studied [1]–[4] under the framework of network utility maximization.

This approach to network resource allocation often can be used to decompose the problem into several sub problems, each of which are amenable to distributed solution. This so-called *optimization decomposition framework* has yielded a rich set of control schemes and protocols, whose architectural implications are discussed in detail in [5]. For example, there is a strong connection between the so-called *primal solution* to the utility maximization problem and TCP-Reno [6], [7] characterized in [8], [9]. Similarly, one can obtain connections between TCP-Vegas and the *dual solution* of the problem [10]. The same approach has been taken in the design of several new protocols such as Scalable TCP [11], [12] (that allows scaling of rate increases/decreases based on network characteristics), FAST-TCP [13] (meant for high bandwidth environments), TCP-Illinois [14] (that uses loss and delay signals to attain high throughput), and TRUMP [15] (a multipath protocol with fast convergence properties).

However, there is a growing realization that throughput cannot be considered as the sole value metric. As mentioned above, in applications such as voice calls, the data is rendered useless after a certain delay threshold. Thus, simply ensuring that the link capacity is not exceeded is not sufficient to provide value in this scenario – how do we ensure that the user is not dissatisfied with the quality of service? In many cases the quality of data transfer over a link decreases with load. For example, metrics such as the delay and the jitter experienced by packets as they pass through a queue depend on the total load on the link. Such quality degradation might

Research was funded in part by NSF grants CNS-0904520, CCF-0747363, DTRA grant HDTRA1-09-1-0051, and Qatar Telecom, Doha, Qatar.

also add up over multiple hops. Indeed, the delay experienced by packets in a flow is the sum of the delays experienced over each hop taken.

Once we have a clear conception of quality degradation as a function of link load, we ask the following question: *can we design a simple distributed algorithm for fair resource allocation under which each users' quality is no worse than a value that she or he declares?* We need to redefine the traditional congestion control problem to include individual source preferences. We denote the degradation in quality of link l with load y_l by a convex increasing function $V_l(y_l)$, and assume that degradation in link quality seen by a flow adds up on the links it traverses. In addition, we assume that the quality degradation is inherent to a link, and is identical for all flows sharing the link. Thus, there are no priorities for any particular flows. We now maximize utility in such a way that the total degradation seen by each source r is required to be bounded by a value σ_r . Thus, the modified objective is

$$\begin{aligned} & \max_{x_r \geq 0, y = Rx} \sum_{r \in \mathcal{S}} U_r(x_r) \\ & \text{subject to } \sum_{l \in \mathcal{L}} R_{lr} V_l(y_l) \leq \sigma_r, \end{aligned} \quad (2)$$

where we assume that $\lim_{y \rightarrow c_l} V_l(y) = \infty$. Note that this is a convex optimization problem where the quality degradation on each route is bounded. While some of our objectives might be achieved by existing schemes such as DiffServ [16], they often require complex priority management methods and per-flow information to be maintained at routers. In this paper, our objective is to design a simple distributed control scheme that can achieve this goal without maintaining per-flow information or prioritizing certain packets at intermediate hops. We overview our main contributions below, with details in the sections following.

Main Results

Classical optimization-decomposition techniques usually yield a “source-rate responds to link-price” type of controller [1], [3]–[5], wherein each link’s price increases with the link-load in order to prevent the link-capacity from being exceeded. As the link-price increases, sources cut down their transmission rates, where the aggressiveness of the source controller is determined by its utility function. However, the solution to our problem has remained elusive due to strong coupling between the quality seen at source that requires a hard guarantee, and the *link quality degradation* that depends on the link-loads along its route.

We first present some examples of what the quality degradation functions might look like in Section II, and discuss examples that we use later in the paper. We then proceed to provide a centralized solution to the problem in Section III. We develop two algorithms to this end. A primal algorithm is presented in Section IV. Our main contribution, a dual algorithm is presented in Section V, and stems from the realization that it is possible to decouple the QoS guarantees at sources and link quality degradation using *effective capacity* that is based

on the link-price and user-dissatisfaction. Once we choose an effective capacity for a link, the quality degradation depends solely on this choice, and not on the actual link-load.

Each source declares its *dissatisfaction* to the links it uses based on the difference between the quality it sees and what it requires. The links set a *price* based on the difference between load and *effective capacity*, which in turn depends on link-load and total user dissatisfaction. This decoupling of link-load and effective capacity appears to have the correct properties to allow distributed solution. Finally, sources use route-price (the sum of all link-prices on a route) to determine the source-rate.

We prove that the algorithm is indeed capable of solving our resource allocation problem using Lyapunov techniques [17]. We illustrate the optimality of the solution reached by our distributed dual control scheme in some selected cases by directly solving the optimization problem. We simulate the controller and conduct experiments on realistic topologies in Section VI. We conclude with pointers to future work in Section VII.

II. EXAMPLES OF QUALITY DEGRADATION FUNCTIONS

We first begin with some ideas of link quality degradation with load. We make several assumptions on the properties of link quality degradation functions. Mathematically, one can list them in the following manner. If the total sum-rate on any link is $y_l = \sum_{r \in \mathcal{S}} R_{lr} x_r$ then

- the quality degradation function $V_l(y_l)$ is non-negative and convex increasing in link-load y_l ,
- the total quality degradation seen by flow r is $\sum_{l \in \mathcal{L}} R_{lr} V_l(y_l)$ (i.e., quality degradation sums up over multiple hops), and
- finally, in our deterministic approach to the problem, we have an implicit assumption that the service process at one link does not impact the arrival process at the succeeding link.

While the above assumptions result in mathematical simplicity of our optimization problem, we believe that they provide acceptable models of quality degradations in communication systems with queues. Below, we justify our assumptions with some common examples of quality degradation functions.

A. Average delay in M/M/1 queues

For an M/M/1 queue with arrival rate x and service rate c , the expected waiting time in the queue is $\frac{x}{c(c-x)}$ for a stable queue, that is when $x < c$. In this case, one can write quality degradation function to be the expected waiting time for any packet in the queue. That is,

$$V(x) = \frac{x}{c(c-x)}.$$

We note that the quality degradation function is always positive and increases from 0 to ∞ when x ranges in $[0, c)$. Further,

$$\begin{aligned} V'(x) &= \frac{1}{(c-x)^2} > 0, \\ V''(x) &= \frac{2}{(c-x)^3} > 0. \end{aligned}$$

Hence, the function is convex increasing.

B. Decay-rate of a fluid buffer with on-off service process

Consider a single server queue with constant-rate arrival x and a two-state On-Off service process where on and off times are exponentially distributed with rates μ and λ respectively. When service is on and the buffer is non-empty, it is serviced at a constant rate $R > x$ such that $x < R\frac{\lambda}{\lambda+\mu}$. It can be shown [18] that the probability of buffer exceeding a threshold z is exponentially decreasing and a possible quality degradation function in this case could be the inverse of this decay-rate. One can write down this decay-rate explicitly in terms of the above parameters as

$$V(x) = \left(-\lim_{z \rightarrow \infty} \frac{\log Pr\{L > z\}}{z} \right)^{-1} = \left(\frac{\lambda}{x} - \frac{\mu}{R-x} \right)^{-1}.$$

If we denote $R\frac{\lambda}{\lambda+\mu}$ by c then, one can write

$$V(x) = \frac{x \left(\frac{c}{\lambda} - \frac{x}{\mu+\lambda} \right)}{c-x}.$$

First, we notice that $V(x)$ is always non-negative in the interval $[0, c)$. Second, we see that when x approaches 0, the value $V(x)$ diminishes to zero. Analogously, when x approaches c , the value $V(x)$ tends to infinity. Additionally,

$$V'(x) = \frac{1}{\mu+\lambda} \left(1 + \frac{\mu}{\lambda} \left(\frac{c}{c-x} \right)^2 \right) > 0,$$

$$V''(x) = \frac{2c^2\mu}{\lambda(\mu+\lambda)(c-x)^3} > 0.$$

Hence, one can conclude it is convex increasing. Recent results [19] suggest that under appropriate conditions, even when packets of a flow traverse from one queue to the next in a network, the delay seen in each queue is independent of the others.

III. CENTRALIZED RESOURCE ALLOCATION

We start by developing ideas on how to solve our resource allocation problem in a centralized fashion and creating model networks that we will use as examples to illustrate the performance of different control loops throughout the paper. We recall our optimization problem, repeated here for convenience,

$$\max \sum_{r \in \mathcal{S}} U_r(x_r) \quad (3)$$

subject to the constraints

$$\begin{aligned} y_l &\leq c_l, \quad \forall l \in \mathcal{L} \\ \sum_{l \in \mathcal{L}} R_{lr} V_l(y_l) &\leq \sigma_r, \quad \forall r \in \mathcal{S} \\ x_r &\geq 0, \quad \forall r \in \mathcal{S}. \end{aligned} \quad (4)$$

Consider the case, where utility functions assume unbounded negative values when $x_r = 0$ and quality degradation functions grow unbounded when sum-rate y_l approach c_l . Then, clearly

$x_r > 0$ and $y_l < c_l$ for optimal solution. Let x_r^* be a feasible point and there exist constants $w_r \geq 0$ such that

$$\begin{aligned} U'_r(x_r^*) - \sum_{s \in \mathcal{S}} w_s \sum_{l \in \mathcal{L}} R_{ls} R_{lr} V'_l \left(\sum_{i \in \mathcal{S}} R_{li} x_i^* \right) &= 0, \quad \forall r \in \mathcal{S} \\ w_r \left(\sum_{l \in \mathcal{L}} R_{lr} V_l \left(\sum_{i \in \mathcal{S}} R_{li} x_i^* \right) - \sigma_r \right) &= 0, \quad \forall r \in \mathcal{S}, \end{aligned} \quad (5)$$

then x_r^* is a global maximum and if U_r is strictly concave, then x_r^* is unique global maximum.

We will illustrate by the following examples how our model takes into consideration all of the desired properties of the quality degradation function and how they impact resource allocation with service guarantees.

A. Homogeneous Tandem Network

Consider n flows sharing a simple tandem network of L links where each link has a constant capacity c and identical quality degradation function V_l associated with each link. All the flows originate at the first node and their destination is the final node as shown in Fig. 1.

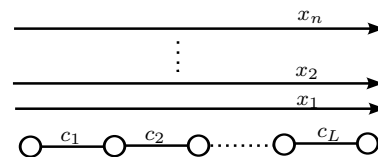


Fig. 1. Tandem network of L links being shared by n flows.

We assume that associated with each flow i is a utility function $a_i \log(x_i)$ corresponding to its throughput x_i , and a service guarantee σ_i . We also take the quality degradation function to be $V_l(y_l) = -\log(1 - \frac{y_l}{c_l})$. We choose this quality degradation function, as it satisfies the desired convexity property. It also captures the effect that for achieving capacity over erroneous communication links, one needs to use arbitrary long codes leading to unbounded variance in available service. Thirdly and very importantly, this choice of quality degradation function gives us analytical expression for x_i 's, that offers valuable insight into trade-off between throughput and service guarantees. Under these assumptions, we have the following optimization problem

$$\max \sum_{i=1}^n a_i \log x_i \quad (6)$$

subject to the constraints

$$\begin{aligned} -L \log \left(1 - \frac{\sum_{i=1}^n x_i}{c} \right) &\leq \sigma_i, \quad i = 1, 2, \dots, n \\ \sum_{i=1}^n x_i &\leq c_l = c, \quad l = 1, 2, \dots, L \\ x_i &\geq 0, \quad i = 1, 2, \dots, n. \end{aligned} \quad (7)$$

The log function ensures that all x_i 's are always positive and also that the sum-rate constraints are never active. Let

w_i be the Lagrange multipliers associated with the quality degradation constraint, then the Lagrangian is given by

$$L(x, w) = \sum_{i=1}^n a_i \log x_i + \sum_{j=1}^n w_j \left(L \log \left(1 - \frac{\sum_{k=1}^n x_k}{c} \right) + \sigma_j \right) \quad (8)$$

Differentiating the Lagrangian with respect to x_i 's and equating them to zero, we obtain

$$c - \sum_{j=1}^n x_j^* = \frac{L \sum_{k=1}^n w_k}{a_i} x_i^*. \quad (9)$$

Defining row vector $\mathbf{1}$ of ones of size n , and $n \times n$ matrix D with real entries such that $D_{jj} = 1 + \frac{L \sum_{k=1}^n w_k}{a_j}$ and $D_{ij} = 1$ for $i \neq j$; we can equivalently write the above equation in the following compact form,

$$x^* D = c \mathbf{1}. \quad (10)$$

Let $i^* = \arg \min \{\sigma_i : i = 1, 2, \dots, n\}$, then $w_i = 0, i \neq i^*$ and $c - \sum_{j=1}^n x_j^* = c \exp(-\sigma_{i^*}/L)$ by KKT conditions. Then,

$$x_i^* = \left(a_i / \sum_{i=1}^n a_i \right) c (1 - \exp(-\sigma_{i^*}/L)).$$

This example verifies that our modeling intuition is right for resource allocation with service guarantees. Here are some observations from this simple example:

- For any finite service requirement, sum-rate is always less than the capacity of each link.
- Throughputs decrease with number of hops due to service requirements.
- When quality degradation is inherent to a link, flow with most stringent service requirements limits the throughput for every other flow.

B. Three-Flows Two-Hop Network

Consider the network in Fig. 2 in which three sources transmit over two links. Let link i have capacity c_i . Assuming log utility and quality degradation functions as in previous example, the resource allocation problem becomes

$$\max \sum_{i=1}^n a_i \log x_i \quad (11)$$

subject to the constraints

$$\begin{aligned} -\log \left(1 - \frac{x_i + x_3}{c_i} \right) &\leq \sigma_i, \quad i = 1, 2 \\ -\sum_{i=1}^2 \log \left(1 - \frac{x_i + x_3}{c_i} \right) &\leq \sigma_3. \end{aligned} \quad (12)$$

Let w_i be the Lagrange multipliers corresponding to quality degradation constraint of flow i , then the *Lagrangian* is written

$$\begin{aligned} L(x, w) = &\sum_{i=1}^3 a_i \log x_i + \sum_{i=1}^2 w_i \left(\log \left(1 - \frac{x_i + x_3}{c_i} \right) + \sigma_i \right) \\ &+ w_3 \left(\sum_{i=1}^2 \log \left(1 - \frac{x_i + x_3}{c_i} \right) + \sigma_3 \right) \end{aligned} \quad (13)$$

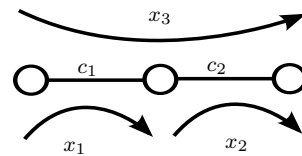


Fig. 2. Three flows sharing a two-link network.

From above equations, we can derive the KKT conditions

$$\begin{aligned} \frac{a_i}{x_i^*} - \frac{w_i + w_3}{c_i - x_i^* - x_3^*} &= 0, \quad i = 1, 2 \\ \frac{a_3}{x_3^*} - \sum_{i=1}^2 \frac{w_i + w_3}{c_i - x_i^* - x_3^*} &= 0, \\ w_i \left(\log \left(1 - \frac{x_i + x_3}{c_i} \right) + \sigma_i \right) &= 0, \quad i = 1, 2 \\ w_3 \left(\sum_{i=1}^2 \log \left(1 - \frac{x_i + x_3}{c_i} \right) + \sigma_3 \right) &= 0. \end{aligned} \quad (14)$$

Let us consider the case when $\sigma_3 < \min\{\sigma_1, \sigma_2\}$. In this case, $w_1 = w_2 = 0$ and $\left(\sum_{i=1}^2 \log \left(1 - \frac{x_i + x_3}{c_i} \right) + \sigma_3 \right) = 0$. For the simple case of $a_i = 1, i = 1, 2, 3$ we have optimal rates

$$x_1^* = x_2^* = 2x_3^* = \frac{2c}{3} (1 - \exp(-\sigma_3/2)). \quad (15)$$

This example verifies our modeling intuition for resource allocation with service guarantees, with same conclusions as in previous example. We will use above two simple examples for numerical studies of our dual algorithm in Section V.

IV. PRIMAL ALGORITHM

We now develop an algorithm that could potentially be used to obtain an approximate solution of our optimization problem. The approach that we use is called the *Primal* method, as it follows from the Primal formulation of the problem. The main idea is to relax the constraints by incorporating them as a cost into the objective. Essentially, the idea is that there is a price to violating the quality constraints, and we maximize utility minus price. Thus we consider the function

$$J(x) = \sum_{r \in \mathcal{S}} (U_r(x_r) - B_r(\sum_{l \in \mathcal{L}} R_{lr} V_l(y_l))), \quad (16)$$

where $B_r(\cdot)$ is a barrier function assumed to be convex increasing, from zero to unbounded values when argument increases from zero to σ_r . To minimize this function, we can use a gradient descent approach, i.e.,

$$\begin{aligned} \dot{x}_r &= k_r(x_r) (U'_r(x_r) - q_r), \\ q_r &= \sum_{s \in \mathcal{S}} B'_s \left(\sum_{l \in \mathcal{L}} R_{ls} V_l(y_l) \right) \sum_{l \in \mathcal{L}} R_{lr} V'_l(y_l). \end{aligned} \quad (17)$$

Since the problem is convex, it is straightforward to show using Lyapunov techniques [2], [9], [17] that the above algorithm converges, and leads to one maximizer of (16). To this end, note that $J(x)$ as defined in (16) is a strictly concave function. We denote its unique maximizer by \hat{x} . Then, $J(\hat{x}) - J(x)$ is non-negative and equals zero only at $x = \hat{x}$. This makes

$W(x) \triangleq J(\hat{x}) - J(x)$, a natural candidate Lyapunov function and we use it in the following proposition, which has a similar proof to that of [9].

Proposition 1. *Consider a network in which all sources follow the primal control algorithm (17). Let $J(x)$ be as defined in (16) and functions $U_r(\cdot), k_r(\cdot), V_l(\cdot)$ and $B_s(\cdot)$ be such that $W(x)$ grows unbounded as $\|x\| \rightarrow \infty$, and $\hat{x}_i > 0$ for all i . Then, the controller in (17) is globally asymptotically stable and the equilibrium value maximizes (16).*

Proof: Differentiating $W(x)$ with time t , we get

$$\begin{aligned} \dot{W} &= - \sum_{r \in \mathcal{S}} \frac{\partial J}{\partial x_r} \dot{x}_r \\ &= - \sum_{r \in \mathcal{S}} k_r(x_r) (U'_r(x_r) - q_r)^2 < 0, \quad \forall x \neq \hat{x}, \end{aligned}$$

and $\dot{W} = 0$ for $x = \hat{x}$. Here, second line of the equation follows from equations (16) and (17). Thus, all the conditions of the Lyapunov theorem are satisfied and we have proved that the system state converges to \hat{x} . ■

However, primal controller suffers from the following handicaps. The approach is not optimal since the relaxation would yield an acceptable solution only if the barrier values at the optimal solution of our original objective (3) were small. Further, the above formulation would not allow for optimal points on the boundary of constraint set. We now approach the problem from a dual perspective to see if we can obtain any better insight.

V. DUAL ALGORITHM

We start by writing down *Dual* version of our resource allocation problem defined in (3) in the hope it yields insight on how to obtain a distributed method of achieving optimal resource allocation. The Dual problem corresponding to the problem defined in (3) is given by

$$D(w) = \max_{x_s \geq 0} \sum_{s \in \mathcal{S}} U_s(x_s) - w_s \left(\sum_{l \in \mathcal{L}} R_{ls} V_l(y_l) - \sigma_s \right).$$

Let x_r^* be the optimal maximizer, then

$$U'_r(x_r^*) = \sum_{s \in \mathcal{S}} w_s \sum_{l \in \mathcal{L}} R_{ls} R_{lr} V'_l(y_l^*).$$

This gives us a system of equations that needs to be solved to find the optimal x_r^* for each w . However, this is in an implicit form that requires the knowledge of load on every link that a flow traverses. Therefore, this approach is not completely distributed.

Nevertheless, this formulation gives us the hint that instead of link load and link-degradation being dependent on each other directly with load $y = Rx$ and degradation $V(y)$, we could break up their coupling. We do this by introducing a new variable \tilde{y} that we refer to as *effective capacity*. The relaxed version of the resource allocation problem is now

$$\max \sum_{r \in \mathcal{S}} U_r(x_r) \quad (18)$$

subject to the constraints

$$\begin{aligned} \sum_{r \in \mathcal{S}} R_{lr} x_r = y_l &\leq \tilde{y}_l \leq c_l, \quad \forall l \in \mathcal{L} \\ \sum_{l \in \mathcal{L}} R_{lr} V_l(\tilde{y}_l) &\leq \sigma_r, \quad \forall r \in \mathcal{S} \\ x_r &\geq 0, \quad \forall r \in \mathcal{S}. \end{aligned} \quad (19)$$

Assuming that our concave utility and convex quality degradation functions ensure that respectively x_r 's and $(c_l - \tilde{y}_l)$'s are always positive, we can express the Dual problem in terms of positive Lagrange multipliers p_l 's and w_r 's

$$\min_{p, w \geq 0} D(p, w) \quad (20)$$

where $D(p, w)$ is the maximum of the *Lagrangian* $L(x, \tilde{y}, p, w)$ with respect to x, \tilde{y}

$$\begin{aligned} D(p, w) &= \max_{x_s, \tilde{y}_l \geq 0} \sum_{s \in \mathcal{S}} U_s(x_s) - \sum_{l \in \mathcal{L}} p_l \left(\sum_{s \in \mathcal{S}} R_{ls} x_s - \tilde{y}_l \right) \\ &\quad - \sum_{s \in \mathcal{S}} w_s \left(\sum_{l \in \mathcal{L}} R_{ls} V_l(\tilde{y}_l) - \sigma_s \right). \end{aligned}$$

Let x^*, \tilde{y}^* be the maximizers for the above problem for any p, w , then

$$\begin{aligned} U'_r(x_r^*) &= \sum_{l \in \mathcal{L}} R_{lr} p_l, \\ p_l &= V'_l(\tilde{y}_l^*) \sum_{r \in \mathcal{S}} R_{lr} w_r. \end{aligned} \quad (21)$$

Now, we find the partial derivatives of $D(p, w)$ with respect to variables p and w

$$\begin{aligned} \frac{\partial D}{\partial p_l} &= \tilde{y}_l^* - \sum_{s \in \mathcal{S}} R_{ls} x_s^*, \quad l \in \mathcal{L} \\ \frac{\partial D}{\partial w_r} &= \sigma_r - \sum_{l \in \mathcal{L}} R_{lr} V_l(\tilde{y}_l^*) \quad r \in \mathcal{S}. \end{aligned} \quad (22)$$

Then the update equations for solving the Dual minimization of the relaxed problem are

$$\begin{aligned} \dot{p}_l &= h_l(p_l) \left(\sum_{s \in \mathcal{S}} R_{ls} x_s^* - \tilde{y}_l^* \right)_{p_l}^+, \quad l \in \mathcal{L} \\ \dot{w}_r &= k_r(w_r) \left(\sum_{l \in \mathcal{L}} R_{lr} V_l(\tilde{y}_l^*) - \sigma_r \right)_{w_r}^+, \quad r \in \mathcal{S}, \end{aligned} \quad (23)$$

where $h_l(\cdot), k_r(\cdot)$ are positive functions and the notation $(z)_\rho^+$ is used to denote the function

$$(z)_\rho^+ = \begin{cases} z & \rho \geq 0 \\ \max\{z, 0\} & \rho = 0. \end{cases}$$

A. Distributed Algorithm

We can now easily see that the above algorithm is distributed in nature. At any time during evolution of our algorithm, we can treat Lagrange multipliers p_l and w_r as link-price and route-dissatisfaction, respectively. A flow r needs to “pay” link-price p_l for congesting link l if it uses the link (with the route-price being the sum of all such p_l s), and w_r is its end-to-end dissatisfaction under the current system state. The effective capacity of link l is \tilde{y}_l and is decoupled from the actual load on this link $y_l = \sum_{r \in \mathcal{S}} R_{lr} x_r$. We denote the sum of link-prices by $q_r = \sum_{l \in \mathcal{L}} R_{lr} p_l$ for any flow r , and sum of route-dissatisfaction on a link l by $\nu_l = \sum_{r \in \mathcal{S}} R_{lr} w_r$ to yield the total dissatisfaction on that link. Note that such a total implies that there is no need to maintain per-flow information at the link. We denote the effective quality degradation seen by any flow r by $\tilde{\sigma}_r = \sum_{l \in \mathcal{L}} R_{lr} V_l(\tilde{y}_l)$. Notice again that due to decoupling through \tilde{y} , the perceived quality degradation is a function of effective capacity, and not the actual link-load.

The algorithm is illustrated in Fig. 3. Although the diagram is reminiscent of traditional “source-rate responds to link-price” [1]–[4] corresponding to the congestion control problem defined in (1), the system is actually very different. The system may be described as follows:

- Each flow r , as it traverses its route, it accrues the price q_r that it needs to “pay” for using each of the links l . Using this route-price, each source computes a feasible rate

$$x_r = U_r'^{-1}(q_r).$$

Furthermore, each source declares its *dissatisfaction* w_r to the links it uses based on the difference between the quality degradation $\tilde{\sigma}_r$ that it sees and σ_r what it is willing to tolerate. The dissatisfaction is updated using

$$\dot{w}_r = k_r(w_r) (\tilde{\sigma}_r - \sigma_r)_{w_r}^+.$$

- Each link detects the total dissatisfaction ν_l of flows sharing it and computes effective capacity

$$\tilde{y}_l = V_l'^{-1}(p_l/\nu_l)$$

and updates the link price by

$$\dot{p}_l = h_l(p_l) (y_l - \tilde{y}_l)_{p_l}^+.$$

Further, the link ensures that the quality degradation suffered by sharing flows is $V_l(\tilde{y}_l)$ by adding to or dropping part of the total flow as needed.

In summary, along with the two traditional elements of source-rate x_r and link-price p_l , we have two additional control variables: source-dissatisfaction w_r and effective capacity \tilde{y}_l (with link-degradation $V_l(\tilde{y}_l)$) that provide two further dimensions of control that are required for distributed solution.

Now, we show that under reasonable assumptions over $V_l(\cdot)$, the slackness condition of sum-rate being less than or equal to *effective capacity* is always satisfied with equality at equilibrium.

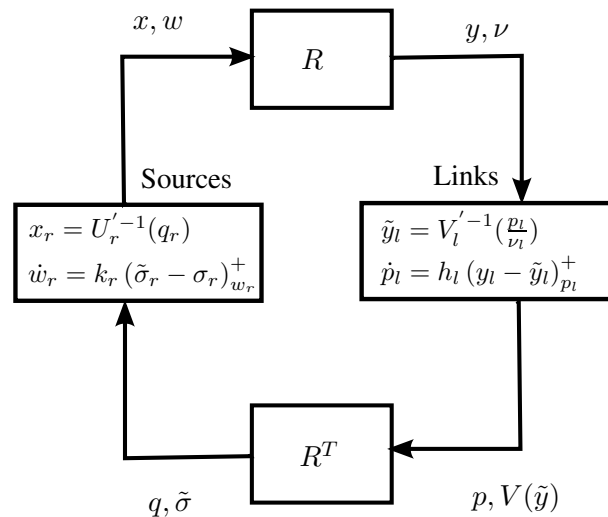


Fig. 3. A block diagram of value-aware resource allocation that allows decoupling of user-dissatisfaction on the source side, and quality on the link side.

Proposition 2. *Let us assume that $V_l(\cdot)$ is strictly convex and increasing. Then, at the equilibrium $y_l = \tilde{y}_l$ for all $l \in \mathcal{L}$.*

Proof: Our proof is by contradiction. Let us assume that there is a $l \in \mathcal{L}$, such that $y_l < \tilde{y}_l$. Then for this l , we have $p_l = 0$. Note that $V_l'(\cdot)$ is a non-negative increasing function. That is, either $V_l'(0) = 0$ or $V_l'(z) > 0$ for all $z \in [0, c_l]$. For the former case, $0 \leq y_l \leq \tilde{y}_l = 0 = V_l'^{-1}(0)$, i.e., $y_l = \tilde{y}_l$. For the latter case, p_l cannot be zero since $V_l'^{-1}(0)$ is not in the feasible range of y_l and hence this will force $y_l = \tilde{y}_l$ at the equilibrium. ■

We have in effect, shown that the equilibrium conditions of our control loop satisfy the KKT conditions of our original optimization problem defined in (3). The conditions are easy to verify, and we may state this result as a corollary of Proposition 2.

Corollary 1. *The stationary point of (23) is a maximizer of the convex optimization problem described by (3).*

B. Global Stability of Distributed Algorithm

It is quite easy to show that the above algorithm is globally asymptotically stable. To show this, we choose our Lyapunov function to be

$$Q(p, w) = D(p, w) - D(\hat{p}, \hat{w}), \quad (24)$$

where \hat{p}, \hat{w} are the unique minimizers of $D(p, w)$. It is clear that $Q(p, w) \geq 0$ for all values of p, w . Also, it is easily seen that $D(p, w)$ grows radially unbounded in p, w for our choice of $V_l(\cdot)$. Therefore, to show that the above algorithm is stable it suffices to show $\dot{D}(p, w) \leq 0$, with equality iff $p = \hat{p}$ and $w = \hat{w}$. Note that at \hat{p}, \hat{w} , one would have $\dot{p}_l = \dot{w}_r = 0$.

Proposition 3. *Let $Q(p, w)$ be as defined in (24) and functions $U_r(\cdot), V_l(\cdot), k_r(\cdot)$ and $h_l(\cdot)$ be such that $Q(p, w)$ grows unbounded with $\|p\|$ and $\|w\|$. Then the controller in (23)*

is globally asymptotically stable and the equilibrium value maximizes (3).

Proof: Differentiating D with respect to time, we get

$$\begin{aligned} \dot{D}(p, w) &= \sum_{l \in \mathcal{L}} \frac{\partial D}{\partial p_l} \dot{p}_l + \sum_{r \in \mathcal{S}} \frac{\partial D}{\partial w_r} \dot{w}_r \\ &= - \sum_{l \in \mathcal{L}} h_l(p_l) (y_l - \tilde{y}_l) (y_l - \tilde{y}_l)_{p_l}^+ \\ &\quad - \sum_{r \in \mathcal{S}} k_r(w_r) (\tilde{\sigma}_r - \sigma_r) (\tilde{\sigma}_r - \sigma_r)_{w_r}^+ \\ &< 0, \quad \forall p, w \neq \hat{p}, \hat{w}, \end{aligned}$$

and $\dot{D}(\hat{p}, \hat{w}) = 0$. Here, the second line of equation follows from equations (22) and (23). Thus, all the conditions of the Lyapunov theorem [17] are satisfied and we have proved that the Lagrange multipliers converge to \hat{p}, \hat{w} . Hence, the system converges to the minimizer of (20). From the convexity of our original problem (3), and Corollary 1, this in turn implies that the stable point is the maximizer of (3). ■

Now, we study our primary examples of homogeneous tandem network and three-flows two-links network to compare our dual algorithm's performance with the optimal solution.

C. Homogeneous Tandem Network

Consider the same setting as in III-A. Then, for an optimal x^*, y^* we would have

$$\begin{aligned} x_i^* &= \frac{a_i}{Lp} \\ y^* &= c - \frac{\sum_{i=1}^n w_i}{p}. \end{aligned}$$

Here, we have assumed that $p_l = p$ and $y_i^* = y^*$ by symmetry. We would also have additional equations from KKT conditions

$$\begin{aligned} p \left(\sum_{i=1}^n x_i^* - y^* \right) &= 0 \\ w_i \left(L \log \left(1 - \frac{y^*}{c} \right) + \sigma_i \right) &= 0. \end{aligned}$$

Finiteness of x_i^* from throughput and quality degradation constraint ensures that $p \neq 0$ and the solution degenerates to the solution of original resource allocation. We can also verify that by simulating dual algorithm on Simulink and using our proposed distributed algorithm to find the optimal operating point as predicted. This exercise also shows that proposed algorithm indeed converges.

We used the following parameters for our Simulink model, $n = 2, L = 3, \sigma_1 = 24, \sigma_2 = 3, c_1 = c_2 = c_3 = 5$ and $a_1 = a_2 = 1$. For this case, we have $x_1^* = x_2^* = 2.5 * (1 - \exp(-1)) = 1.5803$. We have plotted the convergence of source rates with iteration time for both the flows in Fig. 4.

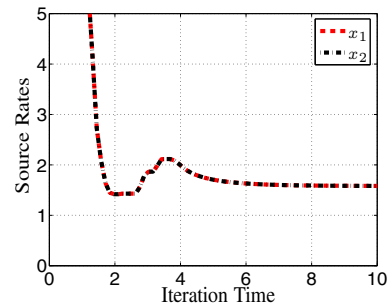


Fig. 4. Distributed resource allocation for Homogeneous Tandem network.

D. Three-Flows Two-Hop Network

Consider the same setting as in III-B. Then, for an optimal x^*, y^* we would have

$$\begin{aligned} x_i^* &= \frac{1}{p_i}, \quad i = 1, 2 & x_3^* &= \frac{1}{p_1 + p_2} \\ y_i^* &= c - \frac{w_i + w_3}{p_i}, \quad i = 1, 2. \end{aligned}$$

Here, we have assumed that $p_l = p$ and $y_i^* = y^*$ by symmetry. We would also have additional equations from KKT conditions

$$\begin{aligned} p \left(\sum_{i=1}^n x_i^* - y^* \right) &= 0 \\ w_i \left(\log \left(1 - \frac{y_i^*}{c} \right) + \sigma_i \right) &= 0, \quad i = 1, 2 \\ w_3 \left(\sum_{i=1}^2 \log \left(1 - \frac{y_i^*}{c} \right) + \sigma_3 \right) &= 0. \end{aligned}$$

Finiteness of x_i^* from throughput and quality degradation constraint ensures that $p \neq 0$ and the solution degenerates to the solution of original resource allocation problem since $y_i^* = x_i^* + x_3^*$, $i = 1, 2$.

For numerical study of convergence of our proposed algorithm for this topology, we used the following parameters in our Simulink model, $\sigma_1 = \sigma_2 = 1, \sigma_3 = 3, c_1 = c_2 = 5$ and $a_i = 1$, $i = 1, 2, 3$. For this case, $\sigma_3 > \sigma_1 + \sigma_2$, and hence we would have $w_3 = 0$. Therefore, $x_1^* = x_2^* = 2x_3^* = \frac{2c_1}{3}(1 - \exp(-\sigma_1)) = 2.1071$. We have plotted the convergence of source rates with iteration time for all flows in Fig. 5.

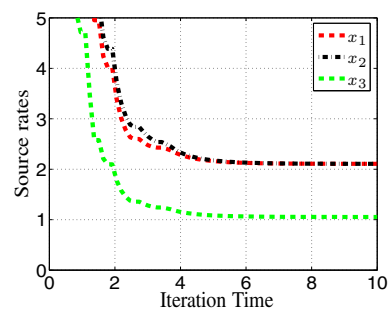


Fig. 5. Distributed resource allocation for three-flow two-hop network.

VI. SIMULATIONS

We utilize two realistic topologies presented in [15] to conduct numerical experiments. Our objective is to study the performance of our value-aware controller in different networking scenarios. We simulated our distributed resource allocation algorithm in Matlab using discrete-time evolution of *link-prices* and *end-to-end dissatisfaction*. Sources send packets at the rate generated by the controller and links average it out by a forgetting factor α . Links base their decision on this average rate.

A. Access-Core Topology

Our first network is an *access-core* topology. It represents a paradigm similar to commercially available Internet access, wherein users have a relatively small access bandwidth (from homes and businesses), connected together by resource rich core-network. User bandwidth is constrained, either directly at the final hop into the home, or at a neighborhood head-end. Applications such as P2P file transfers (low quality constraint), as well as voice and video calls (higher quality constraints) result in end-to-end traffic on such a topology.

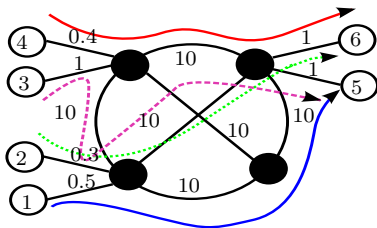


Fig. 6. Access-Core Topology.

We consider the situation when nodes 1 and 3 wish to communicate to node 5 and similarly nodes 2 and 4 to node 6 over an access-core network as shown in Fig. 6. The labels on the links denote their respective capacity. We refer to a flow by its origin node. The QoS constraints on quality of degradation for flows 1-4 are 2, 1, 3, 2, respectively. We plot the convergence of source rates in Fig. 7(a). We also plot load y_l and *effective-capacity* \tilde{y}_l for the diagonal core link used by flows 2 & 3 in Fig. 7(b). We also plotted the quality seen by the flow 2 and its constraint $\sigma_2 = 1$ in Fig. 7(c). Note, we have different rate of convergence of different parameters.

We assumed that core links have a capacity of much higher order than that of access links. Thereby, every link on the core is taken to be of capacity 10, where access node 1 connects to the core with capacity 0.5. Similar capacity for nodes 2-4 are 0.3, 1, 0.4 respectively. We chose nodes 5, 6 to have identical access link capacity of 1.

B. Abilene Topology

Our second network represents the major nodes of the Abilene network topology [20]. The network consists of high bandwidth links, and connects several universities and research labs. Traffic consists of large scale data transfers (low quality

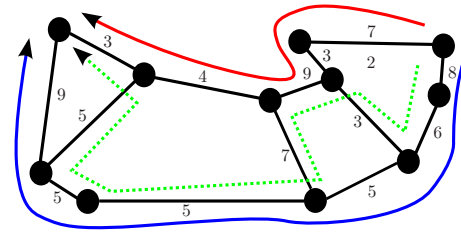


Fig. 8. Abilene Topology.

constraints) and distributed computation (where flows have strict delay constraints).

We consider 3 flows over the Abilene network as shown in Fig. 8 with labels denoting the capacity of the corresponding link. Note, they are of same order. We call the flow on bottom to be flow 1 and one on the top, flow 2. These two flows have QoS constraint on dissatisfaction 8 and 15 respectively. Flow 3 has the zigzag path and has the most stringent QoS constraint of 1. We plot the convergence of flow rates in Fig. 9(a). We also plot load y_l and *effective-capacity* \tilde{y}_l for the link of capacity 3 shared by flows 2 and 3, in Fig. 9(b). We have also plotted the quality seen by the zigzag flow and its acceptable constraint $\sigma_3 = 1$ in Fig. 9(c).

The conclusions that we draw from our simulations are (i) Our value-aware resource allocation algorithm converges to a stable solution, (ii) user quality constraints are satisfied at equilibrium, i.e., the algorithm performs as designed and (iii) the effective capacity is identical to the actual link load at equilibrium showing that our relaxation produces a tight solution.

VII. CONCLUSIONS

In this paper we considered the design of a distributed resource allocation algorithm that would allow each individual flow to specify its measure of value. We assumed that every flow passing through a link suffers a certain quality-degradation due to the load on the link, and that such degradation adds up over the multiple links that the flow traverses. The objective is to ensure that the system throughput is maximized in a fair manner, subject to each flow's quality of service satisfying a hard constraint. Our aim was to ensure that the algorithm should be simple, use local information, and the relays need not maintain per-flow information.

We first showed that attempting to solve this problem by the usual optimization decomposition techniques in Primal formulation yield approximate solutions, and in Dual formulation centralized solutions. However, the observation that decoupling the link-load from the quality degradation using a secondary variable that we call *effective-capacity*, allows us to design such a controller. Under our scheme, the source chooses its rate based on a *route price*, and it declares a *dissatisfaction* based on the quality of service that it sees. Links choose an *effective-capacity* based on dissatisfaction and *link-price*, and modify the price as if the effective-capacity were the actual capacity of the link. The control scheme only requires that links be aware of *aggregate* quantities of the flows using

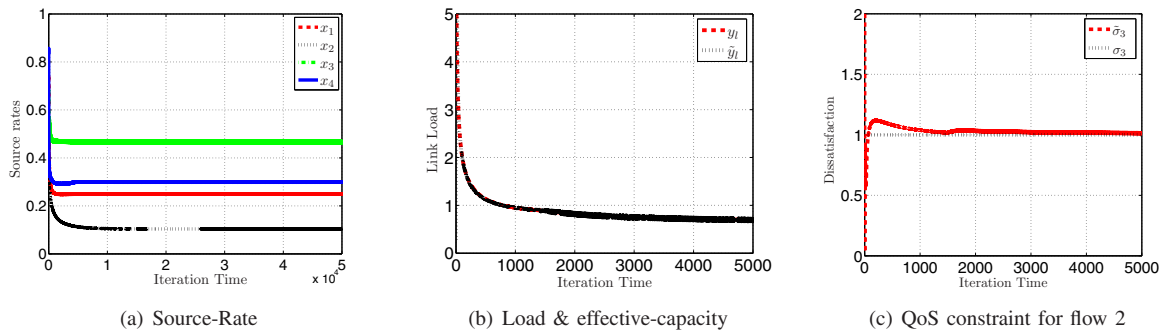


Fig. 7. Convergence performance on Access-core topology.

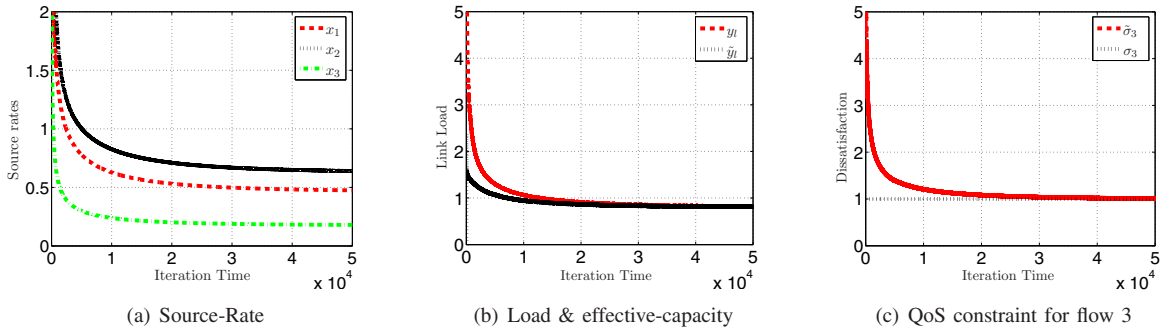


Fig. 9. Convergence performance on Abilene topology.

them, and the sources perform computations solely based on the parameters obtained from the links they traverse, hence satisfying our requirements.

We studied illustrative examples of quality-degradation functions, and certain canonical networks that helped us gain insight into the working of the system. We showed in each case that our distributed controller performs in a near-optimal manner. Finally, we performed simulations on more realistic topologies and illustrated the good performance of our algorithm. In the future we will study protocol development based on our distributed control ideas, which we would then test on a real network.

REFERENCES

- [1] F. P. Kelly, *Models for a self-managed Internet*, Philosophical Transactions of the Royal Society, A358, 2000, pp. 2335–2348.
- [2] F. P. Kelly, *Mathematical modeling of the Internet*, Mathematics Unlimited - 2001 and Beyond (Editors B. Engquist and W. Schmid), Springer-Verlag, Berlin, 2001, pp. 685–702.
- [3] R. Srikant, *The Mathematics of Internet Congestion Control*, Birkhauser, 2004.
- [4] S. Shakkottai and R. Srikant, *Network Optimization and Control*, Foundations and Trends in Networking, Now Publishers, vol. 2, no. 3, 2008.
- [5] M. Chiang and S. H. Low and A. R. Calderbank and J. C. Doyle, *Layering as optimization decomposition: A mathematical theory of network architectures*, Proceedings of IEEE, January, 2007.
- [6] R. Stevens, *TCP/IP Illustrated, Volume 1*, Addison-Wesley, 1994.
- [7] L. L. Peterson and B.S. Davie, *Computer Networks: A Systems Approach*, Morgan-Kaufman, 1999.
- [8] F. P. Kelly, *Charging and rate control for elastic traffic*, European Transactions on Telecommunications, 1997, vol. 8, pp. 33–37.
- [9] F. P. Kelly and A. Maulloo and D. Tan, *Rate control in communication networks: shadow prices, proportional fairness and stability*, Journal of the Operational Research Society, 1998, vol. 49, pp. 237–252.
- [10] S. H. Low and D. E. Lapsley, *Optimization flow control, I: Basic algorithm and convergence*, IEEE/ACM Transactions on Networking, December, 1999, pp. 861–875.
- [11] G. Vinnicombe, *On the stability of networks operating TCP-like congestion control*, Proceedings of the IFAC World Congress, Barcelona, Spain, 2002.
- [12] T. Kelly, *Scalable TCP: Improving Performance in Highspeed Wide Area Networks*, ACM SIGCOMM Computer Communication Review, vol. 33, no. 2, April, 2003, pp. 83–91.
- [13] D. X. Wei and C. Jin and S. H. Low and S. Hegde, *FAST TCP: motivation, architecture, algorithms, performance*, IEEE/ACM Transactions on Networking, December, 2006.
- [14] S. Liu and T. Başar and R. Srikant, *TCP-Illinois: a loss and delay-based congestion control algorithm for high-speed networks*, valuetools '06: Proceedings of the 1st international conference on Performance evaluation methodologies and tools, Pisa, Italy, October, 2006.
- [15] J. He and M. Suchara and J. Rexford and M. Chiang, *Rethinking Internet Traffic Management: From Multiple Decompositions to A Practical Protocol*, in Proc. CoNEXT, New York, NY, December, 2007.
- [16] S. Blake and D. Black and M. Carlson and E. Davies and Z. Wang and W. Weiss, *An Architecture for Differentiated Services*, RFC 2475 (<http://tools.ietf.org/html/rfc2475.txt>) June, 1997.
- [17] H. Khalil, *Nonlinear Systems*, 2nd edition, Prentice Hall, Upper Saddle River, NJ, 1996.
- [18] Lingjia Liu and Parimal Parag and Jia Tang and Wei-Yu Chen and Jean-François Chamberland, *Resource Allocation and Quality of Service Evaluation for Wireless Communication Systems Using Fluid Models*, IEEE Transactions on Information Theory, May, 2007, vol. 53, no. 5, pp. 1767–1777.
- [19] W. Kang and F. P. Kelly and N. H. Lee and R. J. Williams, *State space collapse and diffusion approximation for a network operating under a fair bandwidth sharing policy*, to appear in Annals of Applied Probability.
- [20] Internet2, available at <http://www.internet2.edu/network/>, 2009.