A Unified Approach to Network Design and Control with Non-Cooperative Users

Paper Id: 1928
Number of Pages: 11

Abstract—**A lot of research has *separately* considered non-cooperative congestion control, fairness, pricing and differentiated services for the Internet. However, a solution to one of the problems may not be applicable to others. This leads to implementational complexities and is one of the major reason why the proposed solutions are not implemented in real networks. This paper motivates the need for a *holistic* approach to jointly solve these design problems and proposes a model using an optimization framework for flow control to achieve these objectives. We argue it is possible to handle all of the above problems by using this model, both from the stand points of theoretical analysis and high level implementation issues.**

Keywords— **TCP, Congestion Control, Fairness, Pricing, Service Differentiation, QoS, Optimization**

I. INTRODUCTION

The design and control of computer communication networks has been the focus of numerous studies, through simulations, experiments, as well as theoretical analysis. The design issues can be classified into two parts, those of the network itself and those corresponding to the set of users using the network. Together, they constitute the system. However, the objectives of the users and the network, or between sets of users themselves, can vary widely and can sometimes even be contradictory to each other. Still, a succinct and tractable formulation of the network design and control problem is possible. Simply stated, the general system design problem can be posed as that of optimal resource allocation amongst users given a set of constraints such that the network resource utilization is maximized while maintaining system stability.

The exact definition of “resources” and “utilization” can be thought of as system variables and specific cases of the generalized design problem have been investigated in literature. For example, the system constraints and objectives can be tailored for achieving: a) congestion control, b) appropriate fairness objectives, c) rate differentiation and various forms of Quality of Service (QoS) and d) pricing mechanisms. Though there are bodies of work [16], [17], [24], [8], [31] which address each of these problems individually, no common solution to all of these issues exists.

Solving these problems *separately* opens up an array of implementation issues mainly of integration of these solutions. For example, given two different models, one for congestion control and the other for pricing it becomes an onerous task for the network to provide a single stable solution.

Though these problems appear to be of disparate nature, however, a careful analysis shows that they are tightly intertwined with each other. In this paper, we elaborate on these network design problems, argue that they can be coalesced to solving a single problem and motivate the need for taking the *holistic* view for solving these problems.

We also put forward a framework wherein a *unified* approach can be taken for achieving the heterogeneous objectives mentioned above. We understand that not all issues can be solved by this approach, specifically the issues related to QoS. But we have reasons to believe that the other problems of congestion control, fairness, pricing and rate differentiation can be solved together. In the rest of the paper, we first review the problems concerning Internet design and the solutions articulated for them. Then we introduce and develop a generalized model for solving most of these problems and study the viability of the proposed model with regards to implementation.

II. CURRENT INTERNET DESIGN ISSUES

A significant portion of the success of the current Internet in supporting and deploying numerous applications and protocols owes it to the end-to-end design principle articulated by the authors in [29]. The end-to-end design principle focuses on completeness and correctness of function placement and does not concern itself with the issues regarding congestion control and performance requirements like fairness, pricing and QoS. However, over the years, the sophistication of networks has increased and specific performance and control requirements have arisen. Extensive research has been carried on each of these areas and several different solutions have been proposed for these problems. Each of these issues has been addressed in various degrees [16], [17], [24], [8] and each of these topics has become a field in itself in network research.

However we believe that the most of the problems defined above can be fused to form just a single problem. The historical as well as current approach taken by researchers towards attempting to solve these problems, quite naturally, has been to break them into smaller problems and consider these individually. If this trend is continued, it is very likely that research will concentrate on solving individual issues independently of other factors and thus we lose focus of the overall design problem. As a case in point, we have schemes which attempt to solve the fairness issues but then they leave out the other issues related to rate differentiation and pricing. Thus Internet research has been looking at microscopic problems and we seem to have forgotten the macroscopic objectives.

In this section, we first outline the problems which we believe are most relevant in current and future network design and control problems and have typically been tackled by a bottom-up approach. Leveraging the existing solutions and analysis regarding these problems, a top-down, generalized approach can yield a comprehensive solution which is more practical and viable in terms of implementation.

A. Congestion Control

The congestion control mechanisms used in TCP have been the focus which has undergone a number of enhancements. Now, we have an entire gamut of TCP flavors in the network, starting from TCP Tahoe [16] to TCP Vegas [6] and more being developed using TCP options such as Selective Acknowledgment (SACK) [22] and Explicit Congestion Notification (ECN) [15]. However, these schemes can be differentiated on the basis of way they interpret and react to congestion indications. The reaction to the congestion indications has a profound impact on the equilibrium rate allocation.

The congestion control algorithms being used by the sources also have implications on the fairness and the overall stability of the Internet. Contrary to the perception that the congestion indications a source gets are proportional to the rate allocations, in practice, the congestion indications can be distributed very differently [12], [13], [19]. Furthermore, the fairness also depends on the state of the flow when the source receives the congestion indication. If a TCP source receives a congestion indication when its window is small, the source is very likely to go into timeout allowing the competing sources to grab the available bandwidth. Now since the timed out source will start with a congestion window of 1 packet, the probability it goes into another timeout upon the reception of a congestion indication is higher than the competing sources. This creates problems of fair allocations at the bottleneck.

Recently, congestion control schemes have been evaluated and proposed using optimization frameworks [17], [18], [21], [20]. This framework associates with each user a utility function [30] which measures the user's happiness with respect to the allocated rate. In these papers, the resource allocation problem is proposed as 1) individual users maximize their utility functions and 2) network maximizes every user's utility function given the network capacity constraints.

A significant result of these works is that they show the existence of stable rate adaptation schemes. The authors showed that the equilibrium rate allocation is very closely tied with the utility function the user chooses to maximize. Therein lies our problem. This association of equilibrium rate allocation with the utility function might prompt sources to choose a utility function (and hence congestion control scheme) which yields them higher rate allocations than other competing sources. Such a choice of utility function will still optimize the network and keep it stable, though at the cost of unfair allocations amongst users. This leads us to ask few questions: *Given a means to identify these sources, how can we police these connections? Further, given a definition of compliant rate control schemes like TCP friendliness [11], can we make all the non-compliant sources conform to the defined norms of compliancy?*

We refer to this problem as Non-Cooperative Congestion Control and define "cooperation" as compliant rate control scheme, where the compliance criteria is open and, for instance, could be TCP friendliness. In this paper, we henceforth identify the rate control schemes as either being cooperative (compliant) or non-cooperative (non-compliant) depending on whether they beat other competing sources or compete fairly as per the fairness criteria. In this paper, we loosely interchange cooperative and non-cooperative with compliant and non-compliant respectively.

B. Fairness

Fairness can be defined in a number of ways but its essence in each of these definitions is that it is some measure of the distribution of the allocated rate amongst users. The two most common definitions of fairness are *max-min* [25] and *proportional* fairness [17]. In *max-min* fairness criteria the objective is to maximize the minimum unsatisfied rate allocations. Thus given the same network conditions, two competing flows should get equal share of the bottleneck. On the other hand, in *proportional* fairness the rate allocations are in proportion to the network resources being used. A more general fairness criteria called (p, α) fairness, for a set S of users, can be described as follows

[25]:

$$\sum_{s \in S} p_s \cdot \frac{x_s - x_s^*}{x_s^{*\,\alpha}} \leq 0$$

where x_s is any feasible rate and x_s^* is the *(p, α)* fair rate for the user *s*. The max-min fairness corresponds to *p = 1,* $\alpha \rightarrow \infty$. If the rate allocations are in proportion to the resources used by a user, then such a rate is said to be proportional fair and is defined by *p = 1,* $\alpha = 1$. The reader is referred to [4], [25] for a more thorough study on fairness.

From the discussion in Section II-A the equilibrium rate allocations are decided by the utility functions which the sources choose to maximize. Therefore the distribution of equilibrium rate vector at the bottleneck (or fairness) indirectly becomes a property of the user's in addition to the network and the buffer management scheme it implements.

Consider the following example from [18]. The utility function for TCP Vegas is given by $U_s(x_s) = w_s log(x_s)$ [20] and that for TCP Reno is given by $U_s(x_s) = -w_s x_s^{-1}$ [18], where w_s represents the weight assigned to the flow. Consider a single bottleneck topology where sources are competing against each other. Let there be 50 sources each of TCP Vegas and TCP Reno. Assume that the link capacity to be 300, weights to be 1, the end-to-end propagation delay for both sources to be same. Then the throughput seen by each source can be obtained by solving the following optimization problem:

$$max \sum_{i=1}^{50} log x_i - \sum_{j=51}^{100} \frac{1}{x_j}$$

subject to

$$\sum_{i=1}^{50} x_i + \sum_{j=51}^{100} x_j \leq 300$$

and $x_i, x_j \geq 0 \ \forall i, j$. Solving this problem yields $x_i = 4.0, i \in \{1, ..., 50\}$ and $x_j = 2.0, j \in \{51, ..., 100\}$. Thus, we can see from here the bottleneck is shared between Vegas and Reno in proportion to $\frac{1}{p}$ and $\frac{1}{\sqrt{p}}$ respectively where *p* represents the price or rate of congestion notification. Thus even though the network is fair, the fairness (at the bottleneck) depends on the rate control algorithm chosen by the sources.

The above example has very severe implications. First, the sources can choose rate control schemes which yield higher rate allocations. Another point is that the fairness or the distribution of equilibrium rate allocation depends almost entirely on the rate control algorithms of the sources using the network. Therefore fairness might not be a network prerogative. This prompts another design question: *Can the fairness criteria be decided by the network irrespective of the utility function the sources choose to maximize?*

Scheduling algorithms [9] can achieve the task of disassociating the fairness property from the user's rate control scheme. However, this choice would require placement of schedulers (that achieve the desired fairness) throughout the network. This requires considerable investment in upgrading the network infrastructure. Clearly, this is not a readily deployable solution. Hence we need to look at schemes which can disassociate fairness from user's rate control scheme which require minimal upgrades or more importantly can be easily deployed.

C. Rate Differentiation and QoS

The Internet offers a single class of best-effort service; that is, there is no assurance about rates, delays or other notions of quality including the delivery of the packet. With the introduction and increasing popularity of multimedia and other real-time applications, however, the best-effort services of the Internet may not suffice to satisfy user requirements. Specifically, the different throughput, loss or delay requirement of various applications calls for a network capable of supporting different levels of services, as opposed to a single, best effort level of service.

Delivering a wider variety of services instead of just a single class of best effort service is a fundamental aspect of many of the recently proposed network architectures [2], [5], [32]. DiffServ [2] and RIO (RED with In-profileOut-profile) [8] provide a framework for differentiated services on the Internet. Using the DS byte in packet header sources can indicate their needs for service differentiation as long as the Service Level Agreements have been negotiated between the sources and the network. Then using this information network can assign different congestion indications and thereby differentiating source rates. Other schemes such as Class Based Queuing (CBQ) [14] and its variants have also been proposed for service differentiation.

Rate differentiation techniques can be broadly classified into two groups: Active Queue Management (AQM) based like RIO and Scheduling based [9]. AQM based solutions are hard to implement as the differentiated dropping policies at the bottleneck are often tightly coupled with the arrival process of other sources at the bottleneck. As such however hard one tries no strict or even statistical guarantees for the rate allocation can be given.

Similarly rate differentiation can be provided by scheduling. By allocating different weights to user aggregates the users are assured of a preferred treatment from the network. However, given the dynamic nature of Internet with constant increasedecrease in the number of users and frequent routing changes these weights need to be updated very frequently. Further, weight updating in scheduling algorithm requires some knowledge about traffic patterns. Thus the overheads for rate differentiation with scheduling are very high. This leaves us with a question: *Can rate differentiation be achieved based on utility functions and without resorting to scheduling and bandwidth reservations?*

D. Pricing

As the Internet continues to see high growth rates, providing services and connectivity on the Internet will continue to involve private and public investments. Since no investment is made without hoping for returns, interest in pricing in the Internet was inevitable.

In [10] the author argues “an economy is efficient if it is creating maximum amount of value from the resources available at its command”. Using this argument the author contends that a fully priced Internet should increase the efficiency of use of the network. The author suggests that “transactions amongst users are most efficiently based on capacity per unit time”. Given this background, the author propose a flat fee. for connection to the network based on the bandwidth of the connection. This is generally known as *flat rate pricing* and is the most widely deployed pricing scheme on the Internet wherein unlimited access to the Internet is offered at price, depending on the capacity of the access pipe to the service providers.

However, pricing in the Internet has been proposed more as an incentive for rate differentiation and congestion control than just simple returns. In [23], [24] the authors argue that pricing could help solve the congestion control problem. “As long as the access to bandwidth on the Internet continues to be free, congestion is inevitable”. They cite this as a problem of commons and argue the need for pricing the resources. They contend that when the network is congested, i.e. the resources are scare, pricing needs to be usage-sensitive. Thus, *usage-based pricing* can be used to prioritize usage of a congested resource. Consequently, rate differentiation can be provided by prioritizing and congestion control by pricing.

Lately pricing has received considerable attention and it is widely believed that we will require some kind of usage-based pricing instead of flat-rate pricing regimes. However, contrary to this belief most of the pricing structure on the Internet is flat-rate pricing. Some usage-based pricing structures are also being implemented on the network, but at best they are still in infancy. But most of the usage based pricing, is relegated to just charging in proportion to the connection time. Since this is the easiest pricing scheme to implement it has found a great support in the provider community. However in some networks usage-based is also being implemented according to actual traffic volume.

If the network is going to charge in proportion to the connection duration a malicious user is more likely to get a cheaper service. (This is because he will an disproportionally larger rate allocation than a compliant user.) Thus, *Given the fact that malicious users are going to be present in the network, how can the provider achieve a fair usage based pricing just based on connection duration ?*

Let’s consider another example where the network wants to charge a user on the basis of network resources he uses. (To state mathematically, the provider wants to achieve proportional pricing [17].) Clearly this is not feasible with the current pricing infrastructure implemented on the Internet. Therefore, *If the network were to charge user’s in proportion to the resources being used, how would it achieve this objective ?*

III. A UNIFIED APPROACH TO SOLVING DESIGN GOALS

The Internet has seen phenomenal growth in the last decade in terms of the number of users, traffic volume, number of nodes and the applications and protocols supported. This growth brings with it new design issues and has considerable changed the original objectives. The objectives are moving from “correctness and completeness” [29] to performance issues. As discussed in the previous section, these performance issues present themselves in form of congestion control, fairness, pricing, quality of service, and traffic differentiation etc.

Over the years, considerable research has gone into solving these issues and the research community has benefitted immensely from these solutions. However, surprising as it may seem, most of these solutions have not been adopted by the service providers. The networks still operate with Drop Tail queues and flat rate pricing. Traffic differentiation is still very primitive. Congestion control problem seems to have benefitted most from the research. We now have a more stable network, though as a flip side it has also found out new more aggressive rate control regimes which might harm the existing TCP rate control.

Given the fact that we have solutions to almost every design issue barring QoS (which requires network support), why is it that we are still grappling with implementations? Where did we go wrong, if we did? The answer to this question lies in the way we have tackled these problems. We have tried to isolate these problems and solve them individually, which though fine from an academic/research viewpoint, opens up a Pandora’s box when it comes to integrating these solutions. There lies our problem.

The design issues discussed in the previous section are not independent and are tightly coupled with each other and if we want a "practical" solution, these issues should not be solved in isolation. If we can come up with a robust pricing scheme, then certainly it should be fit for congestion control, fairness and differential service because just by communicating different prices we can achieve these tasks. Similarly if we are concerned about congestion control, we should picture the fairness problem. This fairness problem can then be broadened to encompass traffic differentiation too.

Thus it is safe to say that we need to take a *holistic* view of solving these problems. In this paper we attempt to motivate the need for solving most of the design problems together and provide a framework wherein it might be possible to achieve such a goal. We do not contend that the proposed model is the only framework which achieves the task of integrating the design goals of congestion control, fairness, pricing and rate differentiation. There can be many other such models. However, the primary focus of the paper is to reaffirm the view that an integrated model can solve most of the design issues and should be explored further.

Since the Internet now touches almost every sphere of life, the design choices we make are not exclusive to a small technical community, but together have far reaching implications [30]. Given the fact that we hardly have any working models for most of the new-age design requirements (performance requirements), other avenues which might provide them should be investigated. It is with this hope we investigate a model which can provide a framework for attempting to achieve many of these design objectives.

IV. MODEL

Before we present our framework, we will review some concepts of flow control proposed in [17], [18], [21].

Consider a network of a set $L = \{1, \dots, L\}$ of links, shared by a set $S = \{1, \dots, S\}$ of users. Link $l \in L$ has capacity c_l. User $s \in S$ passes a route L_s consisting a subset of links, $i.e.$, $L_s = \{l \in L \mid s \text{ uses } l\}$. The set of users using a particular link l is given by S_l. When a user s is sending at a rate x_s it achieves a utility $U_s(x_s)$.

Following [21], we model network resource allocation as an optimization problem:

$$max \ \sum_{s \in S} U_s(x_s) \qquad (1)$$

$$st. \ \sum_{s \in S_l} x_s \ \leq \ c_l, \ \ \forall l$$

$$x_s \ \geq \ 0$$

A solution to the above primal problem was provided by considering its Lagrangian dual:

$$min \ D(\mathbf{p}) \qquad (2)$$

$$st. \ \mathbf{p} \geq 0 \qquad (3)$$

where

$$\begin{aligned} D(\mathbf{p}) &= \sum_{s \in S} B_s(p^s) + \sum_{l \in L} p_l c_l & (4) \\ B_s(p^s) &= max_{x_s} U_s(x_s) - x_s p^s & (5) \\ p^s &= \sum_{l \in L_s} p_l & (6) \end{aligned}$$

Using the Karush Kuhn Tucker condition and gradient projection method we get the following source rate and link price updating algorithms

$$\begin{aligned} x_s(t) &= U_s^{'-1}(p^s) & (7) \\ p_l(t+1) &= [p_l(t) + \gamma \cdot (\sum_{s \in S_l} x_s - c_l)]^+. & (8) \end{aligned}$$

A. Remarks

If all users apply the same utility functions, the algorithm in equation (7) will achieve a particular kind of fairness. However, given the same price p^s being communicated by the network, the equilibrium rates can be unequal if users' utility functions are different. Thus even though the network doesn't desire to perceived unfair, a bias in equivalent rates can be created by choosing two different utility functions.

We again visit the example cited in Section II-A. There we showed that given the same network conditions, same end-to-end propagation delays and same prices being communicated to both Reno and Vegas sources, the Vegas source was more likely to have more equilibrium rate allocation. Thus even though the network is fair, the fairness (at the bottleneck) depends on the rate control algorithm chosen by the sources. Additionally it might prompt newer applications (or sources) to choose rate control mechanisms which have higher marginal distribution than that of existing congestion control schemes. This brings to fore new congestion control problem of non-cooperative congestion control which was described in Section II-A.

Several other examples similar to the one mentioned above can be cited for unfair equilibrium allocations. Moreover if the network has so little control over the equilibrium rate allocation, how can it enforce any usage based pricing and rate differentiation? Therefore it makes sense to move the fairness criteria away from the user's rate control scheme to the network. That way the network not only has the flexibility of being fair, but more importantly it can choose the fairness criteria it wants to provide. By doing so, it allows the user's to still have stable congestion control schemes but can solve non-cooperative congestion control by enforcing it's desired fairness criteria. Also, since the network now has indirect control over the equilibrium rate allocation, it can now implement pricing and rate differentiation schemes.

In the next section we describe how the network can actually disassociate the fairness from the user's rate control schemes.

B. Proposed Framework

Assume that the network decides that the final equilibrium rate allocation should be, *as if* every user chose to maximize the same utility function of U_{obj}. Then it follows from equation (7) that the equilibrium rate allocations will be,

$$x_s = U_{obj}^{'-1}(p^s). \tag{9}$$

But we know that the sources change their rate according to equation (7). However, by communicating different prices to the sources we tailor its rate change scheme to match the one defined in equation (9). This can be achieved as follows:

$$x_s = U_s^{'-1}[\,f(p^s)\,]. \tag{10}$$

Thus to make the above rate updating happen we need to communicate a price of $f(p^s)$ instead of p^s to the source s, where

$$f(p^s) = U_s^{'}(U_{obj}^{'-1}(p^s)). \tag{11}$$

Combining equations (10) and (11) we can get equation (9) which is the objective.

A way to understand why this formulation will work is as follows: If the function $f(p^s)$ are strictly non-negative and increasing function in the aggregate price p^s, then $f(p^s)$ can be interpreted as the Lagrangian multiplies solving equation (2). (Please refer to Appendix for the proof that $f(p_l)$ as defined in equation (11) is indeed strictly non-negative and increasing in its argument.) Since the user's utility functions are strictly concave, even this choice of Lagrangian multiplier will maximize user's objective functions with respect to the rate x_s. Alternatively, we can also interpret $f(p_l)$ as the Lagrangian multiplier with respect to the dual formulation in equation (2). Again, $f(p_l)$ are strictly non-negative and increasing functions then the minima will be preserved.

The above explanation is in no way exhaustive and needs to be investigated further. But as stated earlier the primary objective of this paper is to provide a framework and hence at this stage we do not concern ourself with complete proofs of the arguments stated above.

V. DESIGN ISSUES

In this section we elaborate on how we can solve the problems detailed in Section II.

A. Non-Cooperative Congestion Control

The problem of non-cooperative congestion control articulated in Section II-A was that a non-compliant congestion control scheme is more likely to grab an unfair share of bandwidth. As such the need is to be able to identify and police these connections.

The way the network allocates rate at any instant is in proportion to marginal utilities (or $U_s^{'}(x_s)$) of the competing sources. Thus if the marginal utility for a source is always higher than that of the other competing flows, the network is more likely to allocate more bandwidth to it. Since the function we have defined in equation (11) is the marginal utility and moreover is increasing in it's argument, the flow with higher marginal utility will be allocated a higher price. This automatically polices the connections with dis-proportionally higher rate allocations. Similarly for the flows with lower marginal utilities the price is scaled down thus bringing them at par with the other competing sources.

Another related problem addressed in Section II-A was that of making all the congestion control schemes compliant. This task is readily achieved by the re-marking function, $f(p^s)$ defined in equation (11) by

$$\begin{aligned} x_s &= U_s^{'-1}[\,f(p^s)\,] \\ &= U_s^{'-1}[\,U_s^{'}(U_{obj}^{'-1}(p^s))\,] \\ &= U_{obj}^{'-1}(p^s) \end{aligned} \tag{12}$$

where U_{obj} represents the utility function of the compliant scheme and U_s the utility function of any non-compliant scheme.

B. Fairness

The fairness problem as stated in Section II-B was that it was tightly coupled with the source's utility function and network had little control in deciding the equilibrium rate allocation. However, by re-marking the all the sources (re-marking function for compliant sources is just a mapping to itself) the network can now define the equilibrium rate allocations.

In the previous section we showed how the network was able to make the non-compliant sources conform to the definition of compliancy as defined by the network in equations (12). Thus by choosing the objective function,

U_{obj} the network is able to define the fairness available on the network crisply.

It is interesting to note that the proposed framework is robust with regards to change in number of flows, etc. With the proposed architecture the network just has to choose the objective fairness criteria and re-mark all the flows, without worrying about changes in routing topology or in the number of flows.

C. Rate Differentiation

Rate differentiation in the Internet can be provided by choosing an appropriate set of objective functions. Suppose the network wishes to provide two levels of differentiation, one high priority traffic and the other best effort traffic. In such a case, the provider can choose the utility function, say, of TCP Reno as the objective function for mapping the non-priority traffic to best effort. Similarly, it can define how much preference (in terms of rate) it wants to convey to the priority traffic and appropriately choose a utility function which has a higher marginal utility than TCP Reno.

Now the network can use the these two utility functions to provide rate differentiation. The priority traffic is mapped the utility function with higher marginal utility while the rest of the traffic is mapped to TCP Reno's utility function. An important feature of such a rate differentiation is that network can decide the amount of preference for the priority traffic which is hard to pin-point with the RIO or any other multi-level AQM schemes.

Until now we have discussed providing inter-flow traffic differentiation. The same arguments can be extended to provide intra-flow rate differentiation. Intra-flow traffic differentiation applies to cases where a single flow wants to differentiate between high priority packets and low priority packets.

D. Pricing

The proposed framework accommodates a range of static and dynamic pricing schemes. In this section we discuss these schemes in detail.

D.1 Static Pricing

In Section II-D we posed the problem of implementing usage-based pricing, where usage was defined by the proportion of network resources. There have been two popular approaches to static pricing, either based on connection duration or on traffic volume. In the absence of a framework that protects sources against malicious users, both these schemes fall short. If a malicious user is priced for connection duration, he gets to use the same resource at a much cheaper price. Charging based on traffic volume does not take care of fairness in the network.

Now consider static volume-based pricing within the proposed framework. Assume the network chooses the objective function corresponding to proportionally fair allocation, i.e., *log(x)* [17]. Then usage-based pricing can be implemented by just measuring the rate received over the connection duration. This is in fact a more appropriate implementation of usage-based pricing, as compared to the current implementation. With current implementations, even though the user is charged according to traffic volume, there is no way to ensure that the user is priced in proportion to the network resources he is using.

With a static connection-duration-based pricing, the framework again performs better. Since the network-wide fairness criterion is chosen and ensured by the framework, malicious sources cannot monopolize network resources.

D.2 Dynamic Pricing

Dynamic pricing schemes charge the user based on the current state of the network. Service differentiation can be achieved using different marking policies. In our framework, dynamic pricing can be achieved by charging the user for the marking policy he is provided with. Clearly, a user seeing less marking (more expensive) can get a greater share of network resources. There are interesting consequences of pricing based on the marking policy. Analogy for a similar pricing scheme can be found in Paris-Metro Pricing (PMP) [26].

Within a class of users subscribing to the same marking policy, the *effective price* (price per unit rate) for the network resources increases with increase in number of users. This is because there is no guarantee on the rate allocated to the user; the service only provides a greater share of existing resources for the user with a more expensive marking policy.

In addition to an automatic adjustment to the *effective price* with number of users in the system, the framework ensures the chosen fairness criterion within a given class of users with the same marking policy. This strategy can be viewed as a refinement over PMP [26] where each user class not only gets a particular marking policy, but there is an ensured fairness criterion within the class.

In the presence of less number of users in the system, it should be noted that a user with a friendlier marking policy is probably paying higher for the same unit network resource as compared to a user with a cheaper policy. Depending on how concerned the user is about the price of the resource, he could degrade his subscription to a cheaper policy.

Note that, we have assumed that the price for a marking policy is static in the previous paragraphs. This constraint could also be relaxed and price for a preferential policy could vary with demand for network resources.

VI. IMPLEMENTATION ISSUES

In this section we explain how we can implement the *re-marking function* as defined in equation (11). The re-marking function can be implemented either at the individual links or completely at the network edge. In this section we also discuss the how we can characterize a source's utility function. Since the source's utility function's weight are generally a function of the flow's Round Trip Time (RTT), we also discuss its implications in this section.

A. Re-Marking Function

The *re-marking* function $f(p^s)$ can be implemented at each link. The link price update algorithm can be implemented multi-level AQM. Assume we can aggregate the flows based on their utility functions and use it as our classifier. Now given this classifier, we can use a multi-level AQM (depending upon the classes we want to serve) to convey the new price to the sources (of a particular class). However we need to make sure that if we are recalculating the prices at each link, we should be implementing re-marking functions such that new prices when added up over all links match the price as calculated in equation (11). DS byte [2] can be used for classifying the flows.

On the other hand an edge based marker can take the aggregate link price being communicated to the sources and use the re-marking function to get a new price and communicate it to the source.

Since we re-mark packets to convey different prices to sources to make them conform to the network's objective utility functions, we envision the use of ECN [28]. An important point to note is that the proposed architecture won't work if ECN is not enabled. This is because, if were to convey a lower price to some source, definitely we cannot create ACKs for the lost packets, however with ECN we could have easily re-marked the packet by resetting the ECN bit. In the same way, if we were to convey a higher price to source, we cannot drop the packet for it will create problems at the sink. However, we can convey this price in ECN by simply marking the packet. Since, the ECN is readily available in the current Internet, we don't see lack of ECN support as a problem.

B. Estimation of Utility Functions

Throughout the above discussions we have assumed that somehow we have a way of identifying user's utility function. This is very critical for the whole generalized framework to work. We now explain how this can be done.

In [18], [20] the authors have tried to map the window flow control algorithm to utility functions. In fact if the drop probabilities are small, then the knowledge of increase and decrease parameters is sufficient to characterize a source's utility function. (For example, in Appendix VIII-B we show how utility functions can be approximately calculated for TCP Reno with Drop Tail gateways.) The reader is referred to [20] for an exhaustive mapping of utility functions to various TCP regimes and AQM schemes.

In [27] the authors have proposed a tool, TCP Behavior Inference Tool (TBIT) "to characterize the TCP behavior of a remote web server". We believe we can extend this tool to characterize the increase-decrease of any responsive rate control scheme. Thus, if the TBIT is placed at network edges, we can estimate the utility functions. The architecture proposed above will work only if we can characterize the utility function of the sources. With short lived connections or web transfers we believe we won't have sufficient time for characterization. This is one of the limitations of the model and this area needs to explored further. However, a temporary solution can be trying to estimate the utility function of the web server and if we are able to do so, we should cache this information. Thus the next time we should be able to use this information and map the traffic to its appropriate class (based on utility function) for re-marking.

C. Effect of Different RTT on Re-marking function

Another issue which we need to take care is of RTT. The utility functions are generally coupled not only with rate but also with the RTT. For instance, the utility function for TCP Reno with Drop Tail queues is given by [18]

$$U(x) = \frac{-2}{R^2 \cdot x}$$

where x represents the rate and R the RTT. Thus we need some measurement of RTT too. This requirement becomes a big constraint if we need the remapping function at each link. Since we are concerned about the re-marking in the provider's network only, the largest propagation time between the network edge routers in the provider's network can be considered as the RTT. This will in the worst case slow down the re-marking and therefore the convergence to the optimal point. Also such a solution would no longer be constrained by the necessity to know the flow's RTT.

On the other hand if a complete edge based solution is possible (where the re-marking can be done at the network edges) then RTT can be calculated at the edge. In our characterization phase, when we are trying to ascertain the utility functions, RTT can be measured by a small addition to the TBIT. This is because, in TBIT when we measure the congestion window increase we can explicitly store the time-stamps of the starting packet in the window and that of congestion window increase. The difference of these two time-stamps is the RTT.

Arguments can be made that since there are queuing delays in the network, RTTs cannot be accurately computed and more importantly are time-dependent. We contend that we just need an estimate of the RTT, increase and decrease of RTT will at best slow down the convergence to the optimal point.

D. Remarks

The end-to-end design principle has been the corner stone of the Internet growth supporting the growth of newer applications [29]. However, as the demands for performance requirements have grown over the years end-to-end design principle has been laxed to allow putting functionality on the routers at the edge of the network. “This edge orientation for applications and comparative simplicity within the Internet together have facilitated the creation of new applications ...” [3]. Hence any new framework should try to avoid putting anything *in* the network rather should *push* the framework out to the end-systems or network edges.

In our proposed framework the re-marking function can be implemented completely at the edge. The other modules required for our framework, utility function estimator is also placed at the edge. This makes the framework compliant to the end-to-end principles.

VII. CONCLUSIONS AND FUTURE WORK

In this paper we critique current network design philosophy of solving separate design problems (congestion control, fairness, pricing and differentiated services) in isolation. This leads to various implementational complexities and is one of the major reason why the proposed solutions are not implemented in real networks. We argue that most of these problems are closely related and can be combined to form a single problem. Specifically, we motivate the need for taking a *holistic* view for solving these problems and propose a framework in which these objectives may be realized.

We used the optimization flow control framework to provide a unified model for solving non-cooperative congestion control, fairness, pricing and service differentiation problems for the Internet. We contend that this can be achieved by *re-marking* the price information feedback to end users. We have looked at how to realize the *re-marking* function from theory analysis and have considered implementation issues as well. We are currently exploring complete design and implementation details.

We have assumed that the flows are long lived, which is definitely not always the case with the Internet. As such, looking at the optimization dynamics and stability with short lived flows is worthy of further study. Also, extension of this model to rate control schemes for inelastic utility function needs to be investigated.

VIII. APPENDIX

In this section we show that the re-marking function as proposed in equation (11) is non-negative and increasing in its arguments. Also, we show how we can estimate the utility function of TCP Reno with Drop Tail gateways, using window increase and decrease parameters.

A. Re-marking Function

Claim 1: Given the non-negativity constraint on x_s and p_l and strictly concave utility functions U_s and U_{obj}, the function $f(p_l)$ defined in (11) is non-negative and increasing in it’s argument.

Proof: Note $f(p_l) = U_s^{'}(U_{obj}^{'-1}(p_l))$. Recognizing that $U_{obj}^{'-1}(p_l)$ is just x_s from equation (7), we can rewrite $f(p_l)$ as $f(p_l) = U_s^{'}(x_s(p_l))$. Since $U_s(x_s)$ is increasing and strictly concave in its arguments hence $U_s^{'}(x_s) \geq 0$. Hence, $f_{(}p_l)$ is greater than 0.

Let’s define $g(p_l) = U_{obj}^{'}(p_l)$ and it’s inverse as $F(p_l) = g^{-1}(p_l)$. Therefore,

$$F(g(p_l)) = p_l.$$

Now differentiating both sides with respect to p_l we get,

$$\begin{aligned} F^{'}(g(p_l)) \cdot g^{'}(p_l) &= 1 & (13) \\ F^{'}(g(p_l)) &= \frac{1}{g^{'}(p_l)} & (14) \end{aligned}$$

or

$$(U_{obj}^{'-1} U_{obj}^{'}(p_l))^{'} = \frac{1}{U_{obj}^{''}(p_l)}.$$

Since U_{obj} is a strictly concave function, hence

$$(U_{obj}^{'-1}(\cdot))^{'} < 0. \quad (15)$$

Now, differentiating $f(p_l)$ with respect to p_l we get

$$\begin{aligned} f^{'}(p_l) &= U_s^{''}(U_{obj}^{'-1}(p_l)).(U_{obj}^{'-1}(p_l))^{'} \\ &= U_s^{''}(\cdot)(U_{obj}^{'-1}(\cdot))^{'}. \quad (16) \end{aligned}$$

Since U_s is strictly concave therefore $U_s^{''}(x_s) < 0$ and from equation (15) we conclude that $f^{'}(p_l)$ is greater than 0.

B. Utility Function of TCP Reno with Drop Tail Gateway

The window increase algorithm for TCP Reno is given by:

$$W_{t+R} \leftarrow W_t + \alpha/W_t \;\; if \;\; no\; loss \tag{17}$$

$$W_{t+\delta t} \leftarrow W_t - \beta W_t \;\; if \;\; loss \tag{18}$$

where $\alpha = 1$ and $\beta = 0.5$, *W(t)* represents the window at time *t* and *R* represents the RTT. We assume the RTT to be constant and equal to the end-to-end propagation delay. Then rewrite the above equations in discrete time (in steps of RTT):

$$W(t+1) = (W(t) + \frac{1}{W(t)})(1-p)^{W(t)} - \frac{1}{2} \cdot W(t)(1-(1-p)^{W(t)}) \tag{19}$$

where *p* is the loss probability of a packet. Assuming *p* to be small and after some approximations we get

$$W(t+1) - W(t) = \frac{1}{W(t)} - \frac{1}{2} \cdot pW(t) * W(t) \tag{20}$$

Since the price in Drop Tail queues is the packet dropping (marking) probability, without loss of generality the rate change equation can be written as

$$\frac{dx_s}{dt} = \frac{W(t+1) - W(t)}{R} = U_s'^{-1}(p(t)) \tag{21}$$

where x_s is identified as rate and $x_s = \frac{W(t)}{R}$. Thus the rate can be written as

$$x_s = \frac{1}{R}(\frac{2}{p})^{\frac{1}{2}}.$$

Rewriting this equation, we get

$$p = \frac{2}{(Rx_s)^2}$$

and since

$$p = U_s^{'}(x_s),$$

the utility function can be calculated as

$$U_s(x_s) = \frac{-2}{R^2 x_s}.$$

REFERENCES

[1] D. Bansal and H. Balakrishnan. Binomial Congestion Control Scheme. In *Proc. INFOCOM'00*, Mar 2000.

[2] S. Blake et al. An Architecture for Differentiated Services. *IETF RFC 2475*, Dec 1998.

[3] M. Blumenthal and D. Clark. Rethinking the Design of the Internet: The end to end arguments vs. the brave new world. In *ACM Transactions on Internet Technology*, 2001.

[4] T. Bonald and L. Massoulie. Impact of Fairness on Internet Performance. In *ACM SIGMETRICS'01*, Sept 2001.

[5] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: an Overview. *IETF RFC 1633*, Jun 1994.

[6] L. Brakmo and L. Peterson. TCP Vegas: End to End Congestion Avoidance on a Global Internet. *IEEE Journal on Selected Areas in Communications*, 13(8):1465-1480, Oct 1995.

[7] D. Chiu and R. Jain. Analysis of the Increase/Decrease Algorithms for Congestion Avoidance in Computer Networks. *Journal of Computer Networks and ISDN*, 17(1):1-14, Jun 1989.

[8] D. Clark and W. Fang. Explicit Allocation of Best Effort Packet Delivery Service. *IEEEACM Trans. on Networking*, 6(4):362-373, Aug, 1998.

[9] A. Demers, S. Keshav, S. Shenker. Analysis and Simulation of a Fair Queuing Algorithms. In *Proceedings of the ACM SIGCOMM*, 1989.

[10] G. Faulhaber. Pricing Internet: The Efficient Subsidy. In *Building Information Infrastructure*, B. Kahin, (Ed.), McGraw-Hill, 1992.

[11] S. Floyd and K. Fall. Promoting the Use of End-to-end Congestion Control in the Internet. *IEEE/ACM Trans. on Networking*, 7(4):458-472, 1999.

[12] S. Floyd, and V. Jacobson. On Traffic Phase Effects in Packet-switched Gateways. *ACM SIGCOMM Computer Communication Review* 21(2), Apr 1992.

[13] S. Floyd and V. Jacobson. Random Early Drop Gateways for Congestion Avoidance. *IEEE/ACM Trans. on Networking*, 1(4):397-413, Aug 1993.

[14] S. Floyd, and V. Jacobson. Link-sharing and Resource Management Models for Packet Networks. *IEEE/ACM Trans. on Networking*, 3(4):365-386, Aug 1995.

[15] S. Floyd. TCP and Explicit Congestion Notification. *ACM Computer Communication Review*, vol. 24, pp. 10-23, October 1994.

[16] V. Jacobson. Congestion Avoidance and Control. In *Proc. SIGCOMM'88* , Aug 1988.

[17] F. Kelly, A. Maulloo and D. Tan. Rate Control in Communication Networks: Shadow Prices, Proportional Fairness and Stability. *Journal of the Operational Research Society*, Vol.49, pp. 237-252, 1998.

[18] S. Kunniyur and R. Srikant. End-To-End Congestion Control: Utility Functions, Random Losses and ECN Marks. In *Proc. INFOCOM'00*, Mar 2000.

[19] D. Lin and R. Morris. Dynamics of Random Early Detection. *In Proc. SIGCOMM '97*, Oct 1997.

[20] S. Low. A Duality Model of TCP and Queue Management Algorithms. In *Proc. ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management*, Monterey, CA, Sept 2000.

[21] S. Low and D. Lapsley. Optimization Flow Control, I: Basic Algorithm and Convergence. *IEEE/ACM Trans. on Networking*, 7(6):861-875, Dec 1999.

[22] M. Mathis, J. Mahdavi, S. Floyd and A. Romanow. TCP Selective Acknowledgment Options. *RFC 2018*, April 1996.

[23] J. MacKie-Mason and H. Varian. Some Economics of the Internet. In *10th Michigan Public Utility Conference*, Mar, 1993.

[24] J. MacKie-Mason and H. Varian. Pricing the Internet. In *Public access to the Internet*, B. Kahin and J. Keller, (Ed.), Prentice-Hall, 1995.

[25] J. Mo and J. Walrand. Fair End-to-End Window-based Congestion Control. *IEEE/ACM Trans. on Networking*, 8(5):556-567, Oct 2000.

[26] A. Odlyzko. A Modest Proposal for Preventing Internet Congestion. *AT&T Research Labs Tech Report*, 1997.

[27] J. Padhye and S. Floyd. On Inferring TCP Behavior. In *Proc. SIGCOMM'01*, Aug 2001.

[28] K. Ramakrishnan, S. Floyd, and D. Black. The Addition of Explicit Congestion Notification (ECN) to IP. *IETF RFC 3168*, Sept 2001.

[29] J. Saltzer, D. Reed and D. Clark. End-To-End Arguments In System Design. *ACM Trans. on Computer Systems*, 2(4):277-288, Nov 1984.

[30] S. Shenker. Fundamental Design Issues for the Future Internet. *IEEE Journal on Selected Areas in Communications*, 13(7), Sept 1995.

[31] S. Shenker, D. D. Clark, D. Estrin and S. Herzog. Pricing in Computer Networks: Reshaping the Research Agenda. *Telecommunications Policy*, vol 20(3), Reed Elsevier, 1996.

[32] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala. RSVP: A New Resource ReSerVation Protocol. *IEEE Network*, Sept 1993.