On Impact of Non-Conformant Flows on a Network of DropTail Gateways

K. Chandrayana and S. Kalyanaraman
ECSE Dept., Rensselaer Polytechnic Institute

Abstract—**In this paper we evaluate rate distributions between competing flows in a network of DropTail queues. Specifically we look at the case when some of the flows are non- conformant or mis-behaving and it's effect on conformant flows. Our results show in a network of DropTail queues mis-behaving flows can have significantly higher bandwidth allocations at the cost of conformant flows. Further this unequal sharing worsens in a multi-bottleneck scenario where conformant flows may consistently time-out. However the distribution of rates improves if RED is used at the bottleneck thus suggesting deployment of RED.**

In this paper we also look at the fairness from the network's perspective rather then end-user's. As such we propose an analytical model for managing non-conformant or mis-behaving flows by manipulating congestion penalties conveyed to them. We show that this penalty transformation can map a user's utility function, U_s, to any objective utility function, U_{obj}. These penalty transformation modules can be completely implemented at the edge and can also work with Droptail queues. We have analyzed the framework and evaluated it for both single and multi bottleneck scenarios.

I. INTRODUCTION

This paper evaluates the impact of mis-behaving flows on the rate allocations in a network of DropTail queues and proposes an edge-based re-marking framework to manage this misbehavior. Recently congestion control schemes have been evaluated and proposed using optimization frameworks [6], [7], [8]. These framework show that the equilibrium rate allocation is dependent on the utility function the user chooses to maximize. This coupling of equilibrium rate allocation with the utility function might prompt sources to choose a utility function which yields them higher rate allocations. Thus from a network's perspective this creates the problem of "unfair" rate allocations.

Till now the Internet has been running on DropTail queues and though many Active Queue Management (AQM) schemes [5] have been suggested they haven't been deployed for a variety of reasons. Our simulation results show that the problem of unfair rate allocations is specifically severe in a network of DropTail queues. Mis-behaving flows can force conformant flows to consistently timeout, thus grabbing the entire network bandwidth. However, we could avoid shutting out of conformant flows by deploying Random Early Drop (RED) gateways in the network. Though RED gateways can reduce the number of timeouts for a conformant flow, it still does not address the problem of unfair rate allocations. This is because AQM schemes try to manage the congestion and do not differentiate between flows. As such the mis-behaving flows are not penalized sufficiently. In this paper we propose an edge-based re-marking framework to manage mis-behaving flows which can work with both DropTail queues and a network of AQM schemes. However a limitation of our work is that we consider only responsive flows, i.e. flows which react to congestion notification by cutting down their rates.

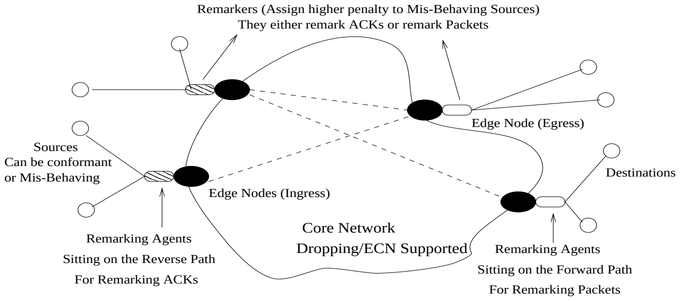


Fig. 1. Model for policing Non-conformant users through Penalty Transformation.

Suppose the network assumes that all the users are maximizing the utility function, U_{obj} and derive their rate control from it. We call, all the users who are actually maximizing U_{obj} as *conformant* users. On the other hand, users who choose to maximize a utility function, U_s i.e. $U_s \neq U_{obj}$ are called *non-ocnformant or mis-behaving* users. In this paper we show through analysis that by transforming the penalty function of the mis-behaving users we can make these sources to behave as if they are conformant sources. Specifically, we propose a modification to the dual formulation [8] to map a user's utility function, U_s, to the network's objective utility function, U_{obj} by conveying a price $U_s'(U_{obj}'^{-1}(p))$ to the non-conformant user. These penalty transformation agents *can be placed on the network edges* and we can choose to re-mark either the packets or acks. Figure 1 shows the model for the remarking framework.

The model presented in this paper can also be thought of as a class of traffic conditioning framework, which can be used to decouple the equilibrium rate allocation of the user and the utility function he chooses to maximize. Also by mapping different sets of flows to a range of target utility function differential service to users can be provided in this framework.

Scheduling algorithms can also achieve the task of disassociating the fairness property from the user's rate control scheme. However, this choice would require placement of schedulers throughout the network. 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 and which require minimal upgrades.

We implemented this framework in NS with the penalty function transformation agents placed in the forward path to re-mark (or drop) the packets. We evaluated it for various single and multi-bottleneck topologies. Our results show that the framework can "re-map" any non-conformant user to co-operative user for any network scenario, if the utility function of the user is known to the network. Further, the framework is robust and works well even in the presence of background web-traffic and reverse-path congestion.

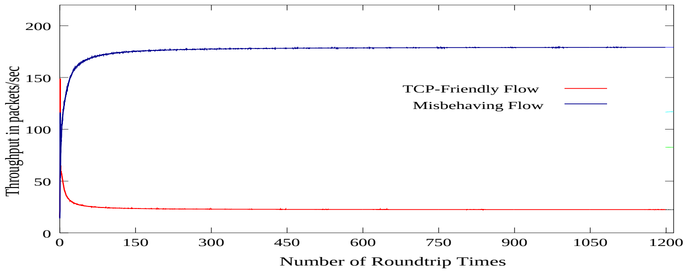
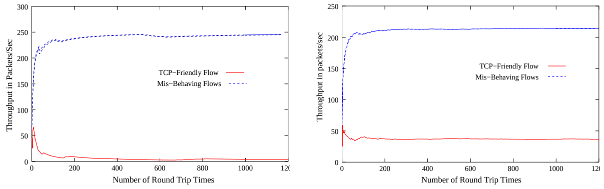


Fig. 2. **Single Bottleneck**: Throughputs (in pkts/sec) for two competing flows, one is TCP Friendly while the other is Mis-behaving.



(a)

(b)

Fig. 3. **Multi-bottleneck**: Throughputs (in pkts/sec) for two competing flows, one is TCP Friendly while the other is Non-conformant with and without Re-Marking.

The rest of the paper is arranged as follows. In Section II we first describe conformance and then show the impact of non-conformant flows on a network of droptail queues. In Section III we present the re-marking model for managing the non-conformant flows. We present the simulation setup in Section IV, results in Section V and discussions on merits and drawbacks of the scheme in Section VI. Finally we present the conclusions and future work in Section VII.

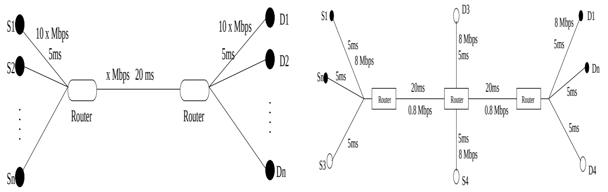
II. NON CONFORMANT FLOWS AND DROP TAIL GATEWAYS

In this section we show the impact non-conformant flows have on rate allocations in a network of DropTail queues. But before we begin our discussion we first relate rate control schemes to utility functions and use it to define and generate non-conformant flows. Let *x* represent the rate. Then we could identify the utility function of any increase/decrease based rate (or window) control scheme with the following relationship

$$U'(x) = \frac{1}{Rxf(x)g(x)} \;:\; f(x) \geq 0,\; 0 \leq g(x) \leq 1 \qquad (1)$$

Further the increase policy, I, and decrease policy, D, of such a scheme can be identified as $I : \frac{1}{f(x)}, D : g(x)$. Binomial Congestion Control Scheme (BCCS) proposed in [1] is one special case of the above model and is given as $I : \alpha/x^k, D : \beta x^l$ where α, β, k, l define the Binomial Algorithm. Because of the simplicity of implementation and understanding, for this work we used BCCS to generate non-conformant flows. In this paper we fixed the values of α, β as 1 and 0.5 respectively. The utility function for a binomial scheme can be approximately calculated as $U_s(x_s) = \frac{-1}{x^{k+l}}$.

In this paper *we have defined flows which are TCP Friendly* [3], *i.e.* k+l = 1 *as conformant*. In this case, the *non-conformant flows are defined by* $k+l < 1$. This is because network allocates more resources to flows which have higher marginal utility, U'_s. Henceforth, in this paper, we will use the *k* and *l* values to identify non-conformant flows. However we need to point out that TCP Friendliness is just one special case of conformance in this framework as the network may choose any other model of conformance as described in detail in our techreport [2].



(a): Single Bottleneck

(b): Multi-Bottleneck

Fig. 4. Topologies used in the Simulations.

In figures 2 and 3 we plot the throughputs for flows competing on a single and multi-bottleneck topologies respectively. We first present the result with a single bottleneck (4 a) of 0.8Mbps and access links of 8Mbps for 2 competing flows. One of the flows is TCP-Friendly while the other is misbehaving flow (k=0, l=0.5). Both the flows have same RTT of 60ms. It can be seen from the figure 2 that in absence of re-marking the non-conformant flow gets most of the bottleneck share. Moreover it beats the TCP-Friendly flow comprehensively.

Figure 4 b) show a multi-bottleneck topology with a TCP-Friendly flow traversing both the bottlenecks while one short mis-behaving flow (k=0, l=0.5), each going through one bottleneck. It can be seen from figure 3 a) TCP-Friendly is almost shut out by the mis-behaving flows, who now get all the bandwidth. Not only is the TCP-Friendly flow is forced into multiple timeouts (23 for this case) but these timeouts occur with very small windows and are often back to back. Similar results were obtained with a higher multiplexing (of flows) but due to space constraints are not reported here. In summary, with DropTail queues mis-behaving flows may get significant share of the bandwidth, almost to the extent of shutting out conformant flows.

Figures 3 b) also plot the throughput when instead of DropTail queues we used RED queues at the bottleneck. (The reader is referred to Section IV for RED settings.) It can be concluded from the figures that though RED improves the shares of TCP-Friendly flow, the unfair rate allocations because of misbehavior of flows persist. This is because the final rate allocations are dependent on the utility function used by the user's and as such an different choices of utility funciton can cause unfair sharing of the bottleneck. Now we outline our re-marking framework through which we can re-map any utility function, U_S to a network's target utility function U_{obj} and thus ensure fair sharing of bottleneck, even with DropTail queues.

III. RE-MARKING FRAMEWORK FOR MANAGING NON-CONFORMANT FLOWS

Consider a user *s*, who is described with the help of his rate, x_s, a utility function U_s and the Set of links which he uses, *L(s)*. Let the network be identified with links *l* of capacity C_l and the set of users using a link, *l*, be given by *S(l)*. Further, assume that the rates are bounded and that the utility functions are *increasing* with rates and *strictly concave*. Then the flow optimization problem is defined as [8]:

$$maximize \sum_{s \in S} U_s(x_s) \qquad (2)$$

$$\textit{subject to} \quad \sum_{s \in S(l)} x_s \;\leq\; C_l, \quad \forall l \qquad (3)$$

for all $x_s \geq 0$. The solution to this problem is given by the following update rules

$$x_s(t) \;=\; U_s^{'-1}\Big(\sum_l p_l\Big) \qquad (4)$$

$$p_l(t+1) \;=\; \Big[p_l(t) + \gamma\Big(\sum_{s \in S(l)} x_s - C_l\Big)\Big]^+ \qquad (5)$$

where p_l are the dual variables of the problem and can be identified as penalties, price or link loss probability [8], [7], [6].

Assume, that the network decides that the final equilibrium rate allocation should be, as if every user chose to maximize a utility function of U_{obj}. Now, if we communicate a link price $f(p_l)$, instead of p_l, then the user-rate updation algorithm is

$$x_s(t) = U_s^{'-1}\Big(\sum_{l \in L(s)} f(p_l)\Big)$$

Further, if we choose $f(p_l) : f(p_l) \geq 0, \forall p_l$, *f(0) = 0* and the following condition holds true

$$\sum_{l \in L(s)} f(p_l) = U_s^{'}\Big(U_{obj}^{'-1}\Big(\sum_{l \in L(s)} p_l\Big)\Big) = g\Big(\sum_{l \in L(s)} p_l\Big) \qquad (6)$$

the the user's rate adaptation will appear as

$$x_s = U_s^{'-1}\Big(\sum_l f(p_l)\Big) = U_{obj}^{'-1}(p_l) \qquad (7)$$

which suggests that the user's now seems to be maximizing a utility function of U_{obj} (instead of U_s. In other words, it shows that we can map the utility function U_s to U_{obj}. Now consider the following update rules

$$p_l(t+1) \;=\; \Big[p_l(t) + \gamma\Big(\sum_{s \in S(l)} x_s - C_l\Big)\Big]^+ \qquad (8)$$

$$x_s(t+1) \;=\; U_s^{'-1}\Big(\sum_l f(p_l(t))\Big) \qquad (9)$$

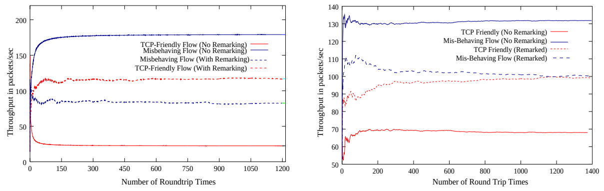
Proposition 1: Given the non-negativity constraint on x_s and strictly concave utility functions U_s and U_{obj}, the update rule defined in equation (8,9) maximizes a flow optimization problem where every user has a utility function of U_{obj}.
Proof: See Appendix VII.

The above link price update rule shows that the price being communicated to the user can be updated at the edge. We now state the algorithm for the edge re-marker as
Edge Marker's Algorithm:

- For each source, receive from the network the total price for the source's traffic as $p^s(t) = \sum_{l \in S(l)} p_l(t)$.
- Recalculate (or Re-mark) the new price for the source as

$$p^s_{new} = \sum_{l \in S(l)} p_l^{new}(t) = g\Big(\sum_{l \in S(l)} p_l(t)\Big).$$

- Communicate this *re-marked* price, p^s_{new} to the source.



(a): DropTail Queues (b): RED Gateways

Fig. 5. **Single Bottleneck**: Throughputs (in pkts/sec) for 2 competing flows on a network of DropTail and RED queues with and without Re-Marking. One flow is TCP Friendly while the other is Non-Conformant (k=0,l=0.5).

IV. IMPLEMENTATION

We implemented the edge based re-marker in the NS (Network Simulator). The edge based re-marker was placed on the forward path and re-marked (or dropped) the packets. The edge re-marker also estimated the loss rate for each flow and subsequently used it for re-marking. For the purposes of estimating losses, we used Exponential Weighted Moving Average (EWMA) and the Weighted Average Loss Indication (WALI) methods of Equation-based rate control algorithm [4]. We updated these loss indications every RTT and we have assumed that the network knows the RTT of the flows. We also assumed that we know the utility functions of all the flows. In this paper we present the results for EWMA based loss-estimator. Similar results were obtained with WALI based estimator. For EWMA based system we gave 60% weight to the history, while with the WALI based estimator we measured samples over 8 windows to estimate losses. A more detailed discussion on the merits and demerits of these schemes can be found in [4].

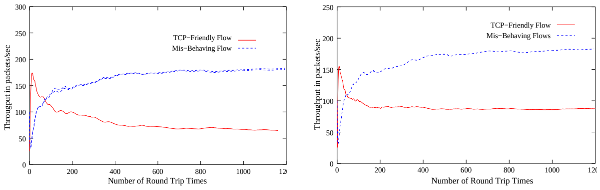
For our simulation we used the congestion control and loss recovery mechanisms of TCP New Reno. The maximum advertised window is set sufficiently high so that it does not constrain the actual window. The simulation time for each setup was 1500 seconds and the packet size was 500B. We plot the throughput of competing flows in packets/sec, averaged over 20 RTT. We assumed that all the flows have infinite data to transfer. The RED queues were setup with min thresh and max thresh set as buffer/3 and 0.8*buffer respectively, where buffer is the total bottleneck buffer length. For the RED queues the dropping probability was set to 0.1 .

V. SIMULATION RESULTS

In the following section we present the results of the simulation for single bottleneck and multiple bottleneck topologies. We evaluate both these setups for different bottleneck link capacity and different type of non-conformant flows. We also validate the robustness and correctness of the scheme with background and cross traffic. We first present the results for single bottleneck topology and follow it with other results in subsequent sections.

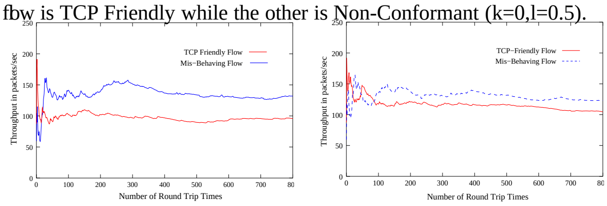
A. Single Bottleneck

We present the result with a single bottleneck of 0.8Mbps and access links of 8Mbps for 2 competing flows. This is the same setup as discussed in Section II. We sampled the packet-stream at the egress router and placed the re-marker there. The re-marker in this case conveyed the transformed penalties to the mis-behaving flows by dropping its packets. Figure 5 shows the



(a): DropTail Queues (b): RED Gateways

Fig. 6. **Multi Bottleneck**: Throughputs (in pkts/sec) for 2 competing flows on a network of DropTail and RED queues with and without Re-Marking. One flow is TCP Friendly while the other is Non-Conformant (k=0,l=0.5).



(a): No Re-Marking (b): Re-Marking

Fig. 7. **Background Traffic**:Throughputs (in pkts/sec) for 10 competing flows in a single bottleneck topology, where 7 flows are TCP Friendly while the other 3 are Non-Conformant with (k=0, l=0.5) **with 65% noise**.

results of with and without the re-marking framework. It can be seen from the figure 5 that in absence of re-marking the non-conformant flow gets most of the bottleneck share. Moreover it beats the TCP-Friendly flow comprehensively as against the same simulation setup with RED queues (as shown in figure 5 b). However, when we start re-marking the misbehaving flows this bias against the TCP-Friendly is reversed. But, it can be seen from the figure 5 that now TCP-Friendly flow gets a better share of the bottleneck. This is because unlike marking, dropping is a stricter means to convey congestion notification as it can lead to timeouts. As such the misbehaving flow suffers.

Figure 5 a) shows the results for a similar setup but with RED queues. When we do not re-mark the non-conformant flow, it garners more bandwidth than the TCP friendly flow. However, re-marking the non-conformant flow makes the two flows to share the bandwidth equitably. Similar results were obtained with higher order of flow multiplexing but are not reported here for the reasons of space constraints.

B. *Multi Bottleneck*

In Section II we saw that the effect of mis-behavior was more pronounced in the case of multi-bottleneck as the non-conformant flows were trying to shut out the TCP friendly flow. In figure 6 a) and b) we plot the results with re-marking enabled in the network, with DropTail and RED queues respectively. Our results suggests that when re-marking is enabled on a network of DropTail queues we can significantly improve the sharing of the bottleneck. On a network of RED queues with re-marking enabled the results are even more appealing thus pointing to virtues of deploying RED in the network.

C. *Background Traffic*

In this section we evaluate the framework in presence of noise-like mice traffic. HTTP sources were added to the persistent non-Conformant and conformant sources. Each http page sends a single packet request to the destination, which then replies with a file of size which was exponentially distributed with 12 1Kb packets. After a source completes this transfer it waits for a random time, which was exponentially distributed with a mean of 1 second and then repeats the process.

Flow Type	DropTail		RED	
	No-Rem	Rem	No-Rem	Rem
TCP-Frndly	55	85	100	200
Non-Conformant	450	400	400	325

TABLE I

CROSS TRAFFIC: COMPARISON OF THROUGHPUT (PACKETS/SEC) FOR NETWORK WITH DROPTAIL AND RED QUEUES WITH AND WITHOUT RE-MARKING.

We used a single bottleneck topology and used different level of flow multiplexing to evaluate the effect of background traffic on the performance of a droptail queue network with and without re-marking. However we report results for one case where there were 10 persistent and of these, 7 flows were TCP Friendly while the remaining 3 where non-Conformant (k=0,l=0.5). The bottleneck bandwidth for this simulation was 10Mbps and a buffer of size 150 packets. Also in this setup we increased the noise sufficiently high to validate the robustness of the scheme in presence of many flows and noise. Figures 7 a) and 7 b) plot the results for the cases where the noise traffic is 65% (or 80 http sources), i.e. mice traffic occupied 65% of the bandwidth. Figure 7 b)shows the robustness of the scheme when sufficiently high (65%) noise is present in the network and the re-marker still manages to efficiently patrol non-Conformant users.

D. *Cross Traffic*

In this section we present the results for our penalty function transformer when two way traffic is present. We evaluate this scenario with the multi-bottleneck topology, where we have 5 TCP Friendly long flows and 5 non-Conformant (k=0, l=0.5) short flows on each bottleneck. Additionally, on the reverse path, there are 5 TCP Reno flows on each bottleneck. The bottleneck bandwidth for this simulation was 10Mbps and a buffer of size 250 packets. Re-Marking, once again achieves fair sharing of the bottleneck (as shown in Table I). However, it can be seen from the results that DropTail queues perform poorly in comparison to RED queues. This further suggests that deployment of RED will help in improving overall network performance, especially in presence of non-conformant flows.

VI. DISCUSSION

In this paper we have proposed an abstract model for modeling and managing non-conformant flows. We believe this is the first work in this direction and an area which needs to be explored further. In this section we will debate the merits and the limitations of the model.

In this paper we have modeled the utility function of non-conformant flows as strictly increasing and concave. Clearly, this is just one part of the non-conformant flows, for CBR flows, pseudo-concave utility functions etc would complete the non-conformant flow realm. CBR sources could be modeled as saturation functions (as used in control systems). The model proposed in this paper, would still be valid for inelastic utility function provided that they are strictly increasing in their argument.

The scheme proposed in this paper is sensitive to loss and utility function estimation. Techniques for loss estimation have been discussed in detail in [4]. Estimation of utility function also has a significant impact on the performance of the re-marking framework. In [2] we have outlined Linear Minimum Mean Squared Error (Linear MMSE) and Non-Linear MMSE methods for estimating the utility function. Both these techniques work well if there are sufficient number of samples, specifically loss samples. However, if there are fewer loss samples then we would have to weigh the samples appropriately such that we discount the samples where we dont have any or one loss.

Despite these limitations, we believe our work is a first step in modeling and managing non-conformant flows. The incentives for such an approach are clearly high. We have showed in this paper that we can effectively decouple the fairness criteria from the user's rate control algorithm. The solution proposed in this paper can be implemented at the network edges. Further we have a flexibility of choosing either to re-mark the packets or acks. In cases where we do not have access to the packet-stream we can re-mark the ack-stream and achieve the goals of the model. Packeteer boxes, deployed widely on the Internet, already do a similar work by accessing the ack-stream and pacing the acks [9] and work well with even 20,000 flows.

In this paper we have presented only a section of simulation results. We refer the reader to our techreport [2] where we provide results for other interesting scenarios. Specifically, we would like to point out the results of our estimation of utility function and a sensitivity analysis of the re-marking framework.

This framework can also provide more robust and fair usage based and flat rate pricing strategies. The reader is referred to [2] for a more detailed discussion on the achievable pricing strategies within this framework.

VII. CONCLUSIONS AND FUTURE WORK

In this paper we looked at the impact of non-conformant flows (or mis-behaving flows) on a network of Droptail and RED queues. Our results show that on a network of DropTail queues non-conformant flows get a large (unfair) share of the bandwidth. Further on multi-bottleneck scenarios the non-conformant flows can almost shut out the conformant flows by pushing them into timeouts. However, on a network of RED queues though the non-conformant still share the bottleneck unfairly but the conformant flows are not shut out. In other words the mis-behavior has a significant impact on a network of Droptail queues than RED queues thus motivating for use of RED.

In this paper we have also proposed an abstract model for modeling and managing non-conformant flows. Specifically we propose a framework to map a user's utility function, U_s, to any objective utility function, U_{obj}, by manipulating congestion penalties. These penalty transformation agents can be completely implemented at network edges. This framework can be used for decoupling fairness from user's rate control schemes and providing more robust usage and flat rate pricing schemes.

We have analyzed the framework and evaluated it for various single and multi-bottleneck scenarios. The model is robust and works well even in presence of high background (web) traffic and reverse path congestion. We are currently working on ways to estimate the utility function of the sources.

APPENDIX

I. PROOF FOR RE-MAPPING NON-CONFORMANT FLOWS FRAMEWORK

We assume that the Utility functions are continuous, strictly concave and increasing in their arguments. Further the rates are bounded by *I: [m_s, M_s].*

Proposition: Given the non-negativity constraint on x_s and strictly concave utility functions U_s and U_{obj}, the update rule defined in equation (8,9) maximizes a flow optimization problem where every user has a utility function of U_{obj}.

Proof: The re-mapping function, $U_s'(U_{obj}^{'-1}(p))$, can also be explained as the solution to the following set of equations:

$$\begin{aligned} \sum_{s\in S(l)} x_s &\leq C_l, \quad \forall l && (10)\\ p_l(\sum_{s\in S(l)} x_s - C_l) &= 0 && (11)\\ U_s^{'}(x_s) &= g(\sum_{l\in L(s)} p_l) && (12) \end{aligned}$$

and $p, x \geq 0$, which are in turn the KKT conditions for the following strictly concave maximization problem

$$\begin{aligned} \underbrace{max}_{x} \quad \textstyle\sum_{s\in S} \quad & \int_0^{x_s} F(U_s^{'}(y_s))\, dy_s && (13)\\ \sum_{s\in S(l)} x_s \quad &\leq \quad C_l, \quad \forall l, \quad x \geq 0 && (14)\\ F \quad &= \quad \left(U_s^{'}\left(U_{obj}^{'-1}(x_s)\right)\right)^{-1} && (15) \end{aligned}$$

Now differentiating the objective function (as defined in equation (13) twice with respect to x_s we get

$$\frac{\partial^2 \int_0^{x_s} F(U_s^{'}(y_s))\, dy_s}{\partial x_s^2} = F'(U_s'(x_s))\, U_s^{''}(x_s) = U_{obj}^{''}(x_s) \qquad (16)$$

Then using assumption we conclude that the objective function (equation 13) is indeed strictly concave. Further it can be concluded that the optimization problem is presented as if all flows have a utility function of U_{obj}.

REFERENCES

[1] D. Bansal and H. Balakrishnan. Binomial Congestion Control Scheme. *Proc. of IEEE INFOCOM*, Tel-Aviv, Israel, March 2000.

[2] K. Chandrayana and S. Kalyanaraman. An Edge-Based Re-Marking Framework for Managing Non-Conformant Flows in Congested Inter-Networks. RPI ECSE Techrepot, ECSE-NET-2003-1, Feb 2003.

[3] 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.

[4] S. Floyd, etal. Equation-Based Congestion Control for Unicast Applications. In *Proc. of ACM SIGCOMM*, Aug 2000.

[5] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. IEEE/ACM Trans. on Networking, 1(4):397–413, Aug 1993.

[6] F. Kelly, A. Maulloo and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the O.R. Society*, 49 (1998) 237-252.

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

[8] S. H. Low, D. E. Lapsley. Optimization Flow Control, I: Basic Algorithm and Convergence. *IEEE/ACM Trans. on Networking*, 7(6):861-75, 1999

[9] Packeteer. http://www.packeteer.com