RI01001 April, 2001 Subject: Computer Science

IBM Research Report

Eliciting Cooperation from Selfish Users: A Game-Theoretic Approach Towards Congestion Control in Communication Networks

Rahul Garg *garg@ieee.org*
Abhinav Kamra *abhinav@iitd.ernet.in*
Varun Khurana *varun@iitd.ernet.in*

IBM Research Division,
IBM India Research Lab,
Hauz Khas, New Delhi - 110016. INDIA.
Phone: +91-11-6861100, Fax: +91-11-6861555

IBM Research Division
Almaden - Austin - Beijing - Delhi - Haifa - T.J. Watson - Tokyo - Zurich

Eliciting Cooperation from Selfish Users: A Game-Theoretic Approach Towards Congestion Control in Communication Networks

Rahul Garg* garg@ieee.org

Abhinav Kamra† abhinav@iitd.ernet.in

Varun Khurana‡ varun@iitd.ernet.in

Abstract

Most of the end-to-end congestion control schemes are "voluntary" in nature and critically depend on end-user cooperation. We show that in the presence of selfish users, all such schemes will inevitably lead to a congestion collapse. Router and switch mechanisms such as service disciplines and buffer management policies determine the sharing of resources during congestion. We show, using a Game Theoretic approach, that all the currently proposed mechanisms, either encourage the behaviour that leads to congestion ("Evil behaviour") or are oblivious to it.

We propose a sample service discipline called Rate Inverse Scheduling (RIS) that punishes Evil behaviour and rewards Good (congestion avoiding) behaviour in such a way that the resulting "Selfish User's Equilibrium" (Nash Equilibrium) leads to (max-min) fair allocation of resources. We show that RIS solves the problems of excessive congestion due to unresponsive flows, aggressive versions of TCP, multiple parallel connections and is fair to TCP as well.

Keywords: Game Theory, Nash Equilibrium, Stackelberg Equilibrium, Generalized Processor Sharing, GPS, Scheduling, Congestion Control, TCP, Fairness, RIS, DWS.

1 Introduction

The Internet has evolved from a small experimental network in an academic environment to a large commercial network and has carried certain legacies which need to be revisited.

A small academic network is a closely knit community where the users tend to be cooperative and more concerned about the collective welfare of the community. In a large commercial network, users tend to be selfish, competitive and more concerned with their individual welfare. Such behaviours have significant impact on the mechanisms designed to share common resources.

For elastic traffic, the sharing of common resources such as link bandwidth and switch buffers is mainly determined by congestion control techniques. Congestion control in the Internet today is primarily done by end host participation using the Transmission Control Protocol (TCP) [22]. TCP was originally designed as a protocol for reliable end-to-end delivery of traffic between two applications on different hosts [36]. After the congestion collapse of the Internet in the early eighties [33], the TCP protocol was modified to include a congestion avoidance mechanism [22]. This mechanism works as follows. For each connection, TCP slowly increases its sending rate. A packet loss on a connection is taken as a signal of incipient congestion in the

*IBM India Research Lab, Hauz Khas, New Delhi - 110016, INDIA

†Indian Institute of Technology, Delhi, Hauz Khas, New Delhi - 110016, INDIA.

‡Indian Institute of Technology, Delhi, Hauz Khas, New Delhi - 110016, INDIA.

network. Upon a packet loss, TCP "voluntarily" reduces its sending rate on that connection to half. It again slowly increases the sending rate of the connection till the next packet loss.

We call such a congestion control scheme as "voluntary congestion control". TCP could continue to increase its sending rate (or keep it same) even if there are packet losses. The decision to reduce the sending rate is purely voluntary and is taken in good faith so as to avoid any congestion in the network. This protocol avoids congestion because the users are cooperative and most of them stick to the principles of voluntary congestion control. In early days of the Internet, such cooperative, end-to-end, voluntary congestion control scheme worked very well, however, in today's environment there are several reasons why such a scheme may not work so well.

Commercialization: Early Internet was an experimental network, and users did not expect to derive a utility from it. Today the Internet has become a commercial success. The users and organizations connected to it pay for its use and expect to derive a utility commensurate with the amount paid for the service. It is therefore unreasonable to expect users to stick to the principles of voluntary congestion control if they can derive a better utility by other behaviours.

Growth: Early days of the Internet had a handful of users in closely-knit communities. Today's Internet has a base of several hundred thousand users spanning almost the entire Globe. Shall we expect to enforce a law that punishes all the congestion causing behaviours in the Internet?

Competition: Today, several competing businesses are dependent on the Internet for their survival. In any competitive game, the goal of a player is to maximize its individual payoff even if that leads to worse total payoffs. In the absence of any law, should we expect a company to stick to the virtues of "voluntary congestion control" if it knows that it can win over its competition otherwise? For instance, is it reasonable to expect a company in the business of selling IP telephony software to stick to voluntary congestion control if its software can deliver a better voice quality otherwise?

Definition of Evil Behaviour: Is there a consensus on which behaviour avoids congestion and which one doesn't? There are several variants of the TCP protocol itself [10, 29, 2, 32, 5] and each variant gives different performance in different congestion conditions.

Awareness: How many Internet users know which type of traffic behaviour causes congestion and which doesn't? How many application writers know this? Shall we expect an average user to know which applications cause congestion and which don't and then act according to the collective welfare?

Selfishness: Individuals in a large group of unknowns are likely to act in a selfish manner. How many users will stick to a congestion-sensitive IP telephony software when a congestion-insensitive IP telephony software can deliver them significantly better voice quality?

Now what will happen if the users of the Internet slowly abolish the principle of voluntary congestion control? A congestion collapse. Every user will attempt to send traffic at its maximum possible rate but will only be able to get a tiny fraction transferred to its destination. This is a manifestation of a much wider problem referred to as the tragedy of commons [21] in the Economics and Game Theory literature.

What is the solution to the problem of commons? Towards one extreme, there are approaches that call for individual cooperation to voluntarily abolish the Evil behaviour (socially responsible investing [30], etc.) and towards the other extreme, there are approaches that call for enacting laws that ban Evil behaviour (pollution laws, traffic rules, etc.). Economists tend to suggest solutions that give sufficient monetary incentives to discourage the Evil behaviour (pollution taxes [14], traffic sensitive tolls [48] etc.). The solution, in general, is to create incentive structures in the system that discourage Evil behaviour.

In this paper we show (in Section 3), using a game theoretic framework, that scheduling and buffer management policies currently in practice either encourage the Evil (congestion causing) behaviour, or are oblivious to

it. Therefore, in the presence of a critical mass of selfish users, the system will inevitably be drawn towards the undesirable state of a congestion collapse.

We then propose (in Section 4) a packet scheduling algorithm called Rate Inverse Scheduling (RIS). RIS punishes Evil (congestion causing) behaviour and gives incentive to Good (congestion sensitive) behaviour. We show (in Section 5) that for a single link shared by N users, sending traffic at the fair rate is the unique Nash and Stackelberg Equilibrium. In an arbitrary network, if RIS scheduling is used at every link, we show that max-min fair rates [7] constitute a Nash as well as a Stackelberg Equilibrium. Further, under the assumption that the user's utility also depends on the loss rate, we show that max-min fair rate is the unique Nash and Stackelberg Equilibrium.

We argue that if RIS is deployed widely in the Internet, then the "best selfish behaviour" for a user will be to estimate its max-min fair rate and send traffic at its max-min fair rate. In the presence of other selfish users, sending traffic at any other rate will result in a lower net utility (goodput) for any user. Therefore by the virtue of their selfishness and the incentive structure in the network (by the means of RIS scheduling algorithm), the best policy for each user will be to estimate its max-min fair rate and send traffic at that rate.

We next show (in Section 6), using simulations, how different versions of TCP perform on RIS scheduling. In particular we not only show that TCP is well protected from unresponsive CBR sources, but also that the unresponsive CBR sources get punished when they start sending at a rate significantly higher than their max-min fair rate. Similarly, versions of TCP differing in their aggressiveness also get similar performance with RIS scheduling. Therefore, if deployed widely, RIS may solve most of the problems pertaining to congestion control [16]. We conclude and discuss possibilities of future work in Section 7.

2 Related Work

There is a large body of research related to congestion control in packet switching networks. The first TCP congestion control algorithm was proposed in [22]. Since then, TCP has evolved and several variants of TCP have been proposed [10, 29, 2, 32, 4, 5]. All of these are end-to-end voluntary congestion control algorithms which require no participation from intermediate routers.

It was later realized that some congestion feedback from the routers can significantly improve the performance of purely end-to-end congestion control algorithms in the Internet. This led to the development of router mechanisms such as Random Early Detection (RED) [18, 28], Early Packet Discard (EPD) [40], Early Congestion Notification (ECN) [38] and ICMP source quench [33, 37]. All these mechanisms rely on the assumption that the users are cooperative and act towards achieving collective welfare. The concerns about selfish user behaviour, non-compliant with the principles of voluntary congestion control are becoming serious day by day [17, 16, 13, 9, 28, 44].

In the context of ATM networks, there are end-to-end approaches [23, 39] and hop-by-hop approaches [31, 47] for congestion control (see [50] for a survey). End-to-end congestion control algorithms are almost always voluntary in nature and require a feedback from the intermediate switches about the congestion status. The feedback could either be an explicit rate [23] indicating the exact max-min fair rate for the flow, or a single bit [39] indicating whether any of the switches in the path of the flow is congested. The ideal hop-by-hop congestion control algorithms do not admit a packet into the network unless the network has adequate resources to guarantee its delivery to the destination.

Most of the current research on scheduling [15, 35, 6, 20] and buffer management policies [12, 45, 28, 13, 44] has focused on protecting well behaved flows from misbehaving flows. None of the previous work in this category

has talked about punishing the misbehaving flows in such a way that the resulting game-theoretic equilibrium (Nash Equilibrium) results in fair resource allocations.

The use of pricing to carry out traffic management and congestion control (of all sorts [24] !!) has been discussed in the seminal works of Kelly [25]. There are other attempts to extend Kelly's ideas in different ways [27, 49]. However, the pricing, especially of Internet bandwidth, is usually driven by business reasons rather than academic reasons. Users as well as ISPs often prefer simple flat pricing over other complex pricing mechanisms [46].

Game Theory has been applied to communication networks in the domain of routing [34, 26] and capacity provisioning [41]. Game Theory was used to analyze the switch service disciplines in a classic paper by Shenker [42]. Using queueing theory models the author shows how Game Theory can be used to formulate and solve the problems in communication networks. Our approach is very similar to that described in the above reference, except that we assume a continuous fluid-flow model of input traffic rather than a discrete queueing theory model of Poisson arrivals. Since, the fluid flow model is amenable to analysis in the context of arbitrary networks, we are able to extend our results to arbitrary networks.

3 Primitive Network, Selfish Users and Game Theory

Consider a link of capacity C shared by N users. There is a sufficiently large shared buffer, a buffer management policy, and a service discipline to partition the link capacity among the users. Assume that user i sends a constant rate traffic flow at a rate r_i (the input rate). Some of this traffic may be dropped due to buffer overflows. Assume that, in steady state the traffic of user i is delivered at the destination with an average output rate γ_i, ($\gamma_i \leq r_i$). The output rate is a function of sending rate of all the N users, the switch service discipline S, and the buffer management policy B. Mathematically, this is written as $\underline{\gamma} = \underline{\gamma}^{SB}(\underline{r})$, where $\underline{r} = (r_1, r_2, \ldots, r_N)$ denotes the N-dimensional vector of input rates and $\underline{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_N)$ denotes the N-dimensional vector of output rates and $\underline{\gamma}^{SB}()$ is the function dependent on scheduling discipline S and buffer management policy B mapping the vector of input rates to the vector of output rates.

In general a user's utility (or satisfaction) depends on its output rate, loss rate and end-to-end delay. However, for a majority of applications the output rate is the most important factor determining the user's utility. For instance, "fire-hose applications" described in [44] are completely loss tolerant. For streaming media applications, loss tolerance can be obtained using forward error correction techniques [3]. For bulk transfer applications, loss tolerance can be achieved using selective retransmissions [7]. Therefore, for simplicity, we assume that a user's utility is an increasing function of its output rate only. The class of such utility functions U^{γ}, is formally defined as[1]:

1. $U \in U^{\gamma}$ maps a user's output rate γ to a real-valued non-negative utility,

2. $U(\gamma') > U(\gamma)$ iff $\gamma' > \gamma$.

If user i was to act in a selfish manner, it would choose a sending rate r_i which will maximize the user's utility (and hence the output rate), irrespective of the amount of inconvenience caused (loss of utility) to the other users. Now, consider what will happen in such a scenario with different packet service disciplines and buffer management policies.

[1]Later, in Section 5 we also consider the class of utility function $U^{\gamma l}$ where a user's utility is also dependent on its loss rate.

Legend	Scheduling discipline	Buffer management policy
FCFS	First come first serve	Shared buffers, drop tail
WF2Q	Worst case fair weighted fair queueing	Per flow buffers, drop tail [6]
LQD	First come first serve	Longest queue tail drop [45]
DT	First come first serve	Dynamic threshold [12]
RED	First come first serve	Random early drop [18]
FRED	First come first serve	Flow RED [28]
RIS	Rate inverse scheduling	Per flow buffers, drop tail

Table 1: Notations for schedulers and buffer management policies

Consider a link of capacity $C = 10Mbps$ shared by five users sending traffic at constant rate. Figure 1 through 4 show the variation in a user's output rate as a function of the user's input rate for different scheduling disciplines and buffer management policies. Refer to Table 1 for notations.

For FCFS, the output rate of a user (and hence the user's utility) always increases by increasing its input rate. However, the slope of the graph depends on the input rate of other users. In such a case, there is an incentive for each user to increase its sending (input) rate, irrespective of what other users are doing. If each user was to act selfishly, to maximize its own utility, the link will end up becoming heavily congested with each user sending traffic at its maximum possible rate and receiving only a tiny fraction of the traffic sent. This is characterized by the concept of Nash Equilibrium [19].

Definition 1 (Nash Equilibrium) *Let $u_i(\sigma, \underline{\sigma})$ represent the utility of player i when the player adopts the strategy σ and all the other players adopt the strategy $\underline{\sigma}$. A strategy profile $\underline{\sigma}^*$ is a Nash Equilibrium if, for every player i,*

$$u_i(\sigma_i^*, \underline{\sigma}_{-i}^*) \geq u_i(\sigma_i, \underline{\sigma}_{-i}^*),$$

for all $\sigma_i \in S_i$, where S_i is the set of strategies that user i can adopt.

In other words, a Nash Equilibrium is a strategy profile where no user has an incentive to deviate unilaterally from its strategy. If all the users are selfish and non-cooperating they would eventually adopt strategies that constitute a Nash Equilibrium.

Here, the vector of input rates $\underline{r}$ constitutes a strategy profile, and each user's utility $U(\gamma)$ is a monotonically increasing function of its output rate γ. For FCFS, RED, and DT resource management policies, the only Nash Equilibrium is when the input rates approach infinity.

If any user behaviour leading to congestion can be termed as Evil behaviour it is appropriate to say that the FCFS policy encourages Evil behaviour. In a network comprising of multiple nodes and links, selfish user behaviour will lead to worse disasters [16, 17] (similar to the congestion collapse) where input rate of each user will approach their maximum possible and output rates will approach zero. To see this, consider the network and flows shown in Figure 5. Assume that FCFS policy is deployed at every link. Every user will send traffic at the rate of access link, 10 Mbps, and will get a net output rate of less than 100Kbps at the final destination.

Now consider WF2Q. The output rate of a user remains equal to its input rate as long as it is less than or equal to its fair rate. When, the input rate becomes larger than the fair rate, the output rate remains constant at fair rate (C/N). The above is true irrespective of the input rates of other users. In such a scenario, a

selfish user will increase its input rate till fair rate. However, a user has no incentive to increase its input rate beyond the fair rate, nor does it have any incentive to keep its input rate down to the fair rate. Therefore, in a network comprising of multiple nodes and links, when a selfish user neither knows its fair rate, nor the resource management policies employed (FCFS or WFQ), it may simply find it convenient to keep on increasing its input rate much beyond the fair rate. For WF2Q, LQD, and FRED, it seems that any vector of input rates where each user's input rate is more than C/N constitutes a Nash Equilibrium. We say that such policies are oblivious to Evil behaviour.

Observe from Figures 1 to 4 that all the resource management policies (except RIS which will be described in following sections) either encourage Evil behaviour or are oblivious to it.

Introducing different grade of services, as proposed by differentiated services [8, 11], makes the situation much more complex. Even if each differentiated services aggregate is policed and shaped, the competing flows within each aggregate, may still have an incentive to send traffic at high rates. As long as there is no incentive for individual users to curtail their sending rates (such as, in the absence of usage based charging, congestion pricing, or edge policing and explicit punishment to misbehaving users) the same argument should hold.

What is really needed for end-to-end congestion control schemes to be effective even in the presence of selfish users, is a resource management mechanism in the interior of the network (i.e., the traffic police) that punishes the Evil behaviour and rewards Good behaviour, while what is currently present in today's networks is the opposite. In the following section we describe a sample service discipline that achieves the purpose of rewarding the Good and punishing the Evil.

4 The Rate Inverse Scheduling (RIS)

We first describe RIS which is defined for idealized fluid-flow model and later describe packetized RIS, a discrete packet-based approximation of RIS that can be implemented in a real switch or router.

RIS is derived from the generalized processor sharing (GPS) scheduling algorithm [35]. Consider a link of capacity C shared by N users sending traffic as N distinct flows. Let $A_i(t)$ represent the amount of traffic of flow i entering the scheduler buffer in the interval $(0,t]$ and $S_i(t)$ be the amount of traffic of flow i served by the scheduler in the interval $(0,t]$. Define $A_i(0) = 0$ and $S_i(0) = 0$ for all i. Define the backlog of flow i at time $t > 0$ as $B_i(t) = A_i(t) - S_i(t)$. Define the total system backlog at time t as $B(t) = \sum_{i=1}^{N} B_i(t)$. Define the input rate of a flow at the link at time t as $r_i(t) = \frac{dA_i(t)}{dt}$ and define the output rate of flow i at the link at time t as $\gamma_i(t) = \frac{dS_i(t)}{dt}$.

A GPS scheduler [35] on the link can be defined as the unique scheduler satisfying the following properties:

Flow Conservation:

$$B_i(t) \geq 0, \forall i, t \geq 0, \tag{1}$$

Work Conservation:

$$\text{If } B(t) > 0, \text{ then } \sum_{i=1}^{N} \gamma_i(t) = C, \tag{2}$$

GPS Fairness:

$$\text{If } B_i(t) > 0, \text{ then } \forall j : \frac{\gamma_i(t)}{\phi_i} \geq \frac{\gamma_j(t)}{\phi_j}, \tag{3}$$

where ϕ_i is the GPS weight assigned to flow i.

The flow conservation property says that for a flow, the traffic served cannot be more than the traffic arrived. The work conservation property says that if there is a non-zero backlog in the system, then the link is not kept idle. The fairness property says that the output (service) rates of all the backlogged flows will be proportional to their respective GPS weights and that of the non-backlogged flows will be lesser.

GPS assigns constant weights to all the flows. RIS differs from GPS in this regard. In RIS, each bit gets a GPS weight that is inversely proportional to that bit's arrival (input) rate. If the bit at the head of the queue of flow i at time t arrived at time $\tau_i(t)$, then in RIS $\phi_i(t) = \theta_i / r_i(\tau_i(t))$. RIS is formally defined as the unique scheduler satisfying the following properties:

Flow Conservation:

$$B_i(t) \geq 0, \forall i, t \geq 0, \tag{4}$$

Work Conservation:

$$\text{If } B(t) > 0, \text{ then } \sum_{i=1}^{N} \gamma_i(t) = C, \tag{5}$$

RIS Fairness:

$$\text{If } B_i(t) > 0, \text{ then } \forall j : \frac{1}{\theta_i}\gamma_i(t) r_i(\tau_i(t)) \geq \frac{1}{\theta_j}\gamma_j(t) r_j(\tau_j(t)), \tag{6}$$

where θ_i is the RIS weight for flow i. Thus, RIS rewards flows with small rates by assigning them large GPS weights and punishing flows with large rates by assigning them small GPS weights. The RIS weights θ may be set in accordance with the pricing, resource sharing or other administrative policies.

Assume, for simplicity that all the RIS weights are set to 1 and all the sources send traffic at a constant rate. Now, all the output rates will also be constant. From the flow conservation property it follows that $\gamma_i \leq r_i$. The fairness condition can be simplified as follows:

$$\text{if } B_i(t) > 0, \text{then } \gamma_i r_i \geq \gamma_j r_j. \tag{7}$$

It follows that if two flows i and j are backlogged, then

$$\gamma_i r_i = \gamma_j r_j = r^2, \tag{8}$$

where r is a constant.

We now prove some important properties of RIS scheduling. In particular, we show that given the input rate of flows, it is possible to define a fair rate such that the output rate of a flow can be completely described in terms of the fair rate and the flow's input rate. Consider an RIS scheduler servicing N constant rate flows of rates $r_i, i \in [1 \ldots N]$ on a link of capacity C. Define the congestion characteristic function $G(x, \underline{r})$ as:

$$G(x, \underline{r}) = \sum_{i=1}^{N} \min\left(\frac{x^2}{r_i}, r_i\right). \tag{9}$$

Define the fair rate f at a link as:

$$f = \begin{cases} C & \text{if } \sum_{i=1}^{N} r_i \leq C, \\ x : G(x, \underline{r}) = C & \text{if } \sum_{i=1}^{N} r_i > C \end{cases} \tag{10}$$

Theorem 4.1 (Fair Rate) *Fair rate f as defined in Eq 10 is unique and*

$$\gamma_i = \min\left(\frac{f^2}{r_i}, r_i\right). \tag{11}$$

The proof in provided in the Appendix.

Lemma 4.1 *The output rate for any flow i is less than equal to the fair rate ($\gamma_i \leq f$).*

Proof: Since $\gamma_i = \min(f^2/r_i, r_i)$, we have the following: If $r_i \leq f$, then $\gamma_i = r_i \leq f$. If $r_i > f$ then $\gamma_i = f^2/r_i < f$. Therefore $\gamma_i \leq f$. □

Lemma 4.2 *Fair rate f is greater than or equal to C/N.*

Proof: If $\sum_{i=1}^{N} r_i \leq C$ then $f = C \geq C/N$. If $\sum_{i=1}^{N} r_i > C$, then from Lemma 4.1 $f \geq \gamma_i$. Summing over all flows we get $fN \geq \sum_{i=1}^{N} \gamma_i$. But from the flow conservation and work conservation properties of RIS scheduler $\sum_{i=1}^{N} \gamma_i = C$, therefore $f \geq C/N$. □

We say that flow i is contributing to the link congestion iff $r_i > f$. Thus, all the flows not contributing to congestion get an output rate equal to their input rate. All the flows contributing to congestion get penalized in proportion to the degree of congestion (r_i/f) caused by them. This is illustrated by the following corollary.

Corollary 4.1 *If $r_i \leq f$ then $\gamma_i = r_i$ else $\gamma_i = f^2/r_i < f < r_i$.*

This above behaviour is also evident from Figures 3 and 4. With RIS scheduling, the output rate of a flow remains equal to its input rate as long as it is less than the fair rate. However, as the input rate exceeds the fair rate, the output rate begins to decline.

Lemma 4.3 (Flow removal) *If $\sum_{i=1,i\neq j}^{N} r_i > C - \gamma_j$ then removal of flow j and adjustment of link capacity as $C' = C - \gamma_j$ does not change the fair rate.*

Proof: From Theorem 4.1, we know that fair rate f satisfies $G(f, \underline{r}) = C$. We rewrite the above equation in the following form:

$$\sum_{i=1(i\neq j)}^{N} \min\left(\frac{f^2}{r_i}, r_i\right) = C - \gamma_j. \tag{12}$$

Substituting the new capacity C' and the new rate vector $\underline{r}'$ in the above equation to get $G(f, \underline{r}') = C'$. Since $\sum_{i=1,i\neq j}^{N} r_i > C'$, and f is the unique rate satisfying $G(f, \underline{r}') = C'$, the fair rate f remains unaffected by removing flow j. □

Lemma 4.4 *If the input rate of flow j is decreased to r'_j from r_j ($r'_j < r_j$), then either $\gamma'_j \geq \gamma_j$ or $\gamma'_j = r'_j$.*

The proof is provided in the Appendix.

4.1 Packetized Rate Inverse Scheduling (PRIS)

In a switch or a router, the traffic does not arrive as a fluid flow. In practice discrete packets containing chunks of data arrive at discrete time boundaries. Therefore a scheduler that works on discrete packets is needed. The packetized RIS scheduler is derived from RIS in the same way as PGPS is derived from GPS. Therefore the implementation details of PRIS are very similar to those of PGPS except for some changes in equations computing the timestamps. It should be straightforward to adapt PRIS to the simplifications of PGPS like Virtual Clock [52], Self Clocked Fair Queueing [20], WF2Q+ [51], Frame based Fair Queueing (FFQ) [43], etc.

Denote the arrival time of k^{th} packet of flow i as a_i^k, length of k^{th} packet of flow i as L_i^k. We model the k^{th} arrival of flow i as if it was a fluid flow of rate $r_i^k = L_i^k/(a_i^k - a_i^{k-1})$ in the interval $(a_i^{k-1}, a_i^k]$. The rate of arrival of all the bits of the packet is given by r_i^k and therefore this packet gets a GPS weight given by $\phi_i^k = \frac{\theta_i}{r_i^k}$. However, since the service cannot begin before the last bit of packet is received, we assume that all the bits of the packet become eligible for scheduling only at time a_i^k. We assume that there is a hypothetical packet collector before the RIS scheduler which collects all the bits of a packet and gives them to RIS only when they become eligible (see Figure 6).

Now, define the finish time of a packet as the time when the last bit of the packet gets serviced in a hypothetical RIS scheduler with a packet collector as shown in Figure 6. PRIS is defined as the scheduler that schedules packets in increasing order of their finish times.

Most of the PGPS implementations are based on the concept of system virtual time and virtual finish time of packets. The PRIS implementation is derived from these PGPS implementations as described below. The scheduler maintains a virtual time function $v(t)$. Upon a packet arrival, each packet is tagged with a virtual finish time as follows:

$$F_i^k = \max\left(F_i^{k-1}, v(a_i^k)\right) + \frac{L_i^k}{\phi_i^k} = \max\left(F_i^{k-1}, v(a_i^k)\right) + \frac{(L_i^k)^2}{\theta_i(a_i^k - a_i^{k-1})}. \tag{13}$$

The packets are serviced in increasing order of their virtual finish times. To compute the virtual time at any instant, an emulation of hypothetical RIS system of Figure 6 is maintained. When a packet k of flow i arrives, it is tagged with a GPS weight of ϕ_i^k and is also given to the RIS emulation. The emulation computes the virtual finish time of the packet by Eq 13. The rate of change of virtual time with real time is given by $d(v(t))/dt = C/(\sum_{i=1}^{N} \phi_i)$, where ϕ_i is the GPS weight corresponding to the packet of flow i which is currently in service in the RIS emulation.

In real practice traffic arrivals are bursty. Therefore, it is better to use a smoothed arrival rate, instead of instantaneous arrival rates for GPS weight computation. The scheduler PRIS with α smoothing sets $\phi_i^k = \theta_i/\hat{r}_i^k$, where $\hat{r}_i^k = \alpha\hat{r}_i^{k-1} + (1-\alpha)r_i^k$. The value of α is taken such that the half life of smoothing is of the order of one round trip time (R) when packet size of L_{max} is used to send traffic. This gives:

$$\alpha = 2^{-L_{max}/(fR)}, \tag{14}$$

where f is the fair rate of the flow.

5 Properties of RIS

We have seen in Section 3 that policies such as FCFS, RED and DT encourage congestion causing behaviour, while policies like WFQ, LQD and FRED are oblivious to such behaviour. In contrast, RIS punishes congestion causing flows as shown in Theorem 4.1 and Corollary 4.1. All flows contributing to a link's congestion have an

incentive to reduce their input rate as shown in Lemma 4.4. So, even the selfish and non-cooperating users will try to avoid congestion. We formally prove these properties in a Game Theoretic framework, first for a single link and then for an arbitrary network.

5.1 Single Link

The fair rate constitutes the unique Nash Equilibrium for N users sharing a single link using RIS. This is formally illustrated in the following theorem.

Theorem 5.1 *Consider a link of capacity C, shared by N users, using RIS scheduling with unit RIS weights. Then $\forall_i r_i = C/N$ is the unique Nash Equilibrium for the system.*

The proof is provided in the Appendix. The naive selfish users will converge to a Nash Equilibrium. However, in case a sophisticated user had information about other users' utility functions or the behaviour of scheduling and/or queue management policies at the gateways, it could choose a value for its input rate and the other users would equilibrate to a Nash Equilibrium in the $N-1$ user subsystem. The leading user can then choose its input rate based on which of these $N-1$ subsystem Nash equilibria maximizes the leading user's utility. This is formally called a Stackelberg Equilibrium.

Definition 2 (Stackelberg Equilibrium) *A strategy profile $\underline{\sigma}^*$ is a Stackelberg Equilibrium with user 1 leading if:*

1. *it is a Nash Equilibrium for users $2 \ldots N$, i.e. $\forall_{i \in [2,\ldots,N]} u_i(\sigma_1^*, \sigma_i^*, \underline{\sigma}_{-1,-i}^*) \geq u_i(\sigma_i^*, \sigma_i, \underline{\sigma}_{-1,-i}^*), \forall \sigma_i \in S_i$, and,*

2. *the leader's utility is maximized, i.e. $\forall \sigma_1 \in S_1, u_1(\sigma_1^*, \underline{\sigma}_{-1}^*) \geq u_1(\sigma_1, \hat{\underline{\sigma}}_{-1})$, where $\hat{\underline{\sigma}}_{-1}$ is a Nash Equilibrium for users $[2 \ldots N]$ when user 1 adopts the strategy σ_1,*

where S_i is the set of strategies that user i can follow.

The leader's utility at a Stackelberg Equilibrium is never less than that in any other set of input rates. So a sophisticated user may try to drive the system towards one of its Stackelberg Equilibria. This can be avoided if Nash and Stackelberg Equilibria coincide. We now show this to be true for a single link with RIS scheduling.

Theorem 5.2 *Consider a link of capacity C shared by N users, using RIS scheduling with unit weights. Then $\forall_i r_i = C/N$ is the unique Stackelberg Equilibrium for the system.*

The proof is provided in the Appendix. Since RIS has a unique and coinciding Nash and Stackelberg equilibria, a user will benefit most by sending at its fair rate only. So even in the presence of competing, non-cooperative and selfish users, RIS will divide the capacity equally among the users. Similarly, if users are assigned RIS weights, the capacity will be divided in proportion to their weights. Now consider what happens if a user doesn't send traffic at its fair rate?

Definition 3 *Given a user with input rate r, define Nash rate for the remaining users as*

$$\eta(r) = \begin{cases} (C-r)/(N-1) & \text{if } r \leq C/N \\ x \geq 0 : x^2/r + (N-1)x - C = 0 & \text{otherwise.} \end{cases} \quad (15)$$

When user 1 sends at r_1, the Nash rate is the rate that constitutes the Nash Equilibrium for the remaining users. This is formally stated in the following Theorem.

Theorem 5.3 *If a user (say user 1) sends at r_1 then $\forall_{i\in(2,N)} r_i = \eta(r_1)$ is the unique Nash Equilibrium for the remaining $N-1$ users.*

The proof is provided in the Appendix. So if $r_1 < C/N$, then the spare capacity is divided equally among the others. Whereas, if $r_1 > C/N$, user 1 is penalized for sending more than its fair rate and other users can increase their output rate. In either case, user 1 is at a loss for not sending at its fair rate and $\eta(r_1)$ is the best input rate for the other users. This is illustrated by the following Lemma.

Lemma 5.1 *If $r > C/N$ then: (1). $\eta(r) < r$ and (2). $\forall \underline{r} : r_1 = r, f(\underline{r}) \geq \eta(r) > C/N$.*

The proof is provided in the Appendix. This implies that if a user sends at a rate r_1 which is more than the fair rate C/N, it may get penalized (when other $N-1$ users send at rate $\eta(r)$) and get an output rate $\eta^2(r)/r$ which is less than r.

5.2 Arbitrary Network of Links

The results for a single link ensure fair sharing of capacity even in the presence of non-cooperating selfish users. But what happens in case of an arbitrary network, where a number of flows having different paths interact over links of different capacities?

Max-min fairness [7] is a well known notion of fairness in an arbitrary network. Consider an arbitrary network servicing N flows. Denote by $\underline{\mathcal{M}}$, the $1 \times N$ vector of max-min fair rates of these flows through this network. It turns out that even in an arbitrary network of links, max-min fair input rates constitute a Nash as well a Stackelberg Equilibrium if RIS is deployed at each link.

Theorem 5.4 *Consider N users sending their traffic as N distinct flows through an arbitrary network with RIS scheduling at each link. Then the max-min fair rates $\underline{\mathcal{M}}$ constitute a Nash as well as a Stackelberg equilibrium for the users.*

The proof is provided in the Appendix. Besides $\underline{\mathcal{M}}$, there may be other equilibria also, and users may try to affect which equilibrium to reach. But if RIS is used at each link, any other equilibrium will result in packet drops for some flows. This is illustrated in the following Lemma.

Lemma 5.2 *Consider N users sending their traffic as N distinct flows through an arbitrary network with RIS scheduling at each link. $\underline{\mathcal{M}}$ is the unique Nash as well as the Stackelberg equilibrium in which there are no losses in the system.*

The proof is provided in the Appendix. Till now we have assumed that a user's utility depends only on its output rate. In general a user's utility may depend on its loss rate as well. If a user can get the same output rate with two different loss rates, it should prefer the smaller loss rate. We formally define this class of utility functions ($U^{\gamma l}$) as follows:

1. $U \in U^{\gamma l}$ maps a user's output rate γ and loss-rate l to a real-valued non-negative utility.
2. $U(\gamma', l) > U(\gamma, l)$ iff $\gamma' > \gamma$.
3. $U(\gamma, l') < U(\gamma, l)$ iff $l' > l$.

The following theorem induces a stronger result for a network with RIS schedulers with $U^{\gamma l}$ utilities that is similar to Theorem 5.1 and Theorem 5.2 for a single link.

Theorem 5.5 *Consider N users sending their traffic as N distinct flows through an arbitrary network with RIS scheduling at each link. Let U_i be the utility function of user i. If $\forall_i U_i \in U^{\gamma l}$, then the max-min fair rates $\underline{\mathcal{M}}$ is the unique Nash and Stackelberg Equilibrium.*

The proof is provided in the Appendix. Therefore the "best selfish behaviour" for a user is to send traffic at its max-min fair rate.

6 Simulations with TCP Users

The results in the previous section imply that the "best selfish behaviour" for a user in the presence of other selfish users is to send traffic at its max-min fair rate. However, the max-min fair rate depends on (a) the link capacities, (b) the number of flows through each link and (c) the path of each flow. A user will not know these parameters in general and thus will not be able to know its max-min fair rate. In this section we illustrate through simulations that the TCP style Additive Increase Multiple Decrease (AIMD) algorithms are able to stabilize at the max-min fair rate, (and at Nash rates in the presence of misbehaving users), when RIS scheduling is deployed. Moreover, specific versions of TCP and the round trip times of individual flows have little impact on the average output rate of a flow. Therefore, RIS scheduling solves most of the problems of congestion control in the presence of misbehaving users [16, 17].

6.1 Simulation Scenario

The simulation scenario is shown in Figure 7, the bottleneck link has a capacity of 10 Mbps and propagation delay of 2 ms. There are 5 access links of capacity 100 Mbps and propagation delay 30 ms.

The buffer size for a flow at each link was set to its round trip delay- bandwidth product. PRIS with α smoothing, per flow buffers and tail drop was used. NS [1] was used to carry out all the simulations.

6.2 Unresponsive flows

A flow that does not change its input rate during congestion is referred to as an unresponsive flow [16]. Responsive flows back off by reducing their sending rate upon detecting congestion while unresponsive flows continue to inject packets into the network thus grabbing a larger share of the bandwidth. Therefore, the presence of

unresponsive flows gives rise to unfairness in bandwidth allocation. With PRIS scheduling deployed, we show using simulations that even TCP style AIMD algorithms can estimate their max-min fair rate (or Nash rate in the presence of misbehaving flows).

The bottleneck link is shared by 5 users, 4 using (responsive) TCP and one using (unresponsive) CBR who sends traffic at a constant rate. Figure 8 shows the average output rates for a representative TCP user as the input rate of the unresponsive user is varied. Each point in the graph represents a simulation of 20 seconds. However, the output rates correspond to the average rate in the last 10 seconds of the simulation when they get stabilized. The Nash rate as defined by Eq 15 of the remaining 4 users is also shown. Note that two sets of simulations were conducted, one where all 4 TCPs are Tahoe and the other where all 4 are Sack.

Figure 8 shows that the TCP users are able to get close to their Nash rate and penalize the unresponsive user, when it causes congestion, according to Theorem 5.3. Note that TCP Sack is able to grab a slightly larger share than TCP Tahoe. Since there are 4 TCP flows, each TCP is able to get only one fourth of the extra bandwidth left by CBR after it is penalized.

6.3 TCP Versions

Different versions of TCP like Tahoe, Reno [32], Vegas [10], and Sack [29] are known to perform differently [32]. We show that with PRIS scheduling there is very little difference in the output rates achieved by these versions. The simulation scenario is shown in Figure 7 with different versions of TCP at n2, n3, n4 and n5. Figure 9 shows the total bytes of a flow transferred as a function of time. The output rate is given by the slope of the graph. Since the slopes are almost identical we see that all versions get identical rates.

6.4 Multiple vs Single Connection

Opening multiple simultaneous connections is a very simple way to grab more bandwidth from drop-tail gateways. This is illustrated in Figure 10. The simulation scenario is shown in Figure 7 with drop-tail employed at node n0. The users at nodes n2, n3, n4, n5 and n6 open 1, 2, 3, 4 and 5 TCP Reno connections to node n1 respectively. Figure 10 shows the plot of bytes transferred vs time for all the 5 users. It is clearly visible that a user opening more simultaneous connections is able to grab more bandwidth.

However, this is not the case with PRIS scheduling when all the TCP connections of a user are treated as part of a single flow. Figure 11 shows a plot of bytes transferred vs. time when PRIS is deployed at node n0. We see that all users get an almost identical bandwidth.

6.5 TCP with different Round Trip Times

The Round Trip Time (RTT) of a TCP connection determines how fast it adapts itself to the current state of the network. A connection with smaller RTT is able to infer the status of the network earlier than a connection with larger RTT. Therefore, large RTT connections typically achieve lower output rates. This is illustrated in Figure 12.

The simulation scenario is the same as shown in Figure 7 except that the propagation delays of links (n2-n0), (n3-n0), (n4-n0), (n5-n0) and (n6-n0) are 5, 10, 20, 50, 100 ms respectively. The value of α for PRIS was taken to be 0.9 which corresponds to the α of the minimum RTT flow according to Eq 14.

Figure 12 illustrates that larger RTT flows get less bandwidth with drop-tail gateways. Figure 13 shows that when PRIS is deployed, after a few drops initially, all flows are able to achieve almost identical rates.

6.6 Network

In this section we show that with PRIS deployed in a network adaptive flows like TCP are able to estimate and converge to their max-min fair rates. The simulation scenario is shown in Figure 14. There are 6 TCP Reno flows. The paths of flows 0, 1, 2, 3, 4, and 5 are (n0-n2-n1), (n4-n3-n5), (n0-n2-n3-n4), (n1-n2-n3-n5), (n0-n2-n3-n5) and (n1-n2-n3-n4) respectively. The buffer size of each flow on each gateway was taken to be 300 packets which is the bandwidth delay product of the flow with the largest RTT. For this scenario the max-min fair rate [7] for flows 0 and 1 is 5 Mbps and for flows 2, 3, 4, and 5 is 2.5 Mbps.

Figure 15 shows a plot of output rate vs. time for all 6 flows. We see that after some initial fluctuations all flows are able to converge at their max-min fair rates.

7 Discussion, Conclusions and Future Work

Using the techniques of Game Theory, we showed that the current service disciplines and buffer management policies either encourage congestion causing (Evil) behaviour, or are oblivious to it. We proposed a sample scheduler by the name Rate Inverse Scheduling (RIS) and showed that it encourages Good behaviour and punishes Evil behaviour. We showed that for a single link with RIS scheduling, fair rates constitute the unique Nash and Stackelberg Equilibrium. We also showed that for a network with RIS scheduling at every link, the max-min fair rates constitute a Nash as well as a Stackelberg Equilibrium in an arbitrary network with RIS scheduling at every link. With the additional assumption that the users are also sensitive to their packet loss rate, we showed that max-min fair rates constitute the unique Nash and Stackelberg Equilibrium. Therefore, when the RIS scheduling is deployed, even the selfish users will begin to cooperate and will try to estimate their max-min fair rate and send traffic only at that rate. Using simulations we showed that most of the TCP variants are able to estimate their max-min fair rate reasonably, irrespective of their versions and round trip times (RTT). With RIS scheduling, in the presence of unresponsive users who send their traffic at constant bit rate, TCP users are able to extract the penalty, and obtain better average output rates.

None of the ideas presented in this paper are entirely novel. The idea of Game Theory applied to congestion control is motivated by the works of Kelly [24] and is very similar to the works of Shenker [42].

We realize that the biggest limitation of this work is that it requires per-flow queueing and scheduling in the core routers, which may render it useless in a realistic situation. However, this work presents a radically different view of the problem of congestion control in communication networks and gives a sample technique (the RIS scheduling algorithm) that can be used to solve the problem in a Game Theoretic framework. Based on this work, one may be able to design "core stateless" policies [44] with similar properties.

RIS is an instance of generic class of schedulers called the Diminishing Weight Schedulers (DWS). In DWS, the GPS weight assigned to a bit is a decreasing function of the bit's arrival (input) rate. DWS is formally defined as a class of schedulers satisfying the properties of flow conservation, work conservation, and DWS fairness. DWS fairness is same as RIS fairness except that the GPS weights are given as $\phi_i(t) = W_i(r_i(\tau_i(t)))$, where $W(r)$ is a monotonically decreasing function (the diminishing weight function) of r. If the same diminishing weight function is used for all the flows, then the properties of Nash and Stackelberg Equilibrium outlined in Section 5 should hold true. Different diminishing weight functions result in different penalty structures. For instance the function $WI^k(r) = 1/r^k, k > 1$ should impose stricter penalty to the flows contributing to congestion than the rate inverse function. Hence the equivalent Nash rate for $WI^k()$ should be more than that for the rate inverse function $WI^1()$.

This leaves an important question unanswered. It the inverse rate the best diminishing weight function? What is the best diminishing weight function? The answer of this question will, in general, depend on what the end user behaviour is.

This paper also opens up several newer areas of work in the field of game theoretic congestion control. Some versions of TCP give a reasonable performance with RIS. How to characterize TCP performance on RIS? Can there be better end user behaviours other than TCP? What is a good end user behaviour for reliable multicast protocols assuming that RIS scheduling is deployed in the routers?

References

[1] Network simulator (NS), 2001. URL: http://www.isi.edu/nsnam/ns/.

[2] J.-S. Ahn, P. Danzig, Z. Liu, and L. Yan. An evaluation of TCP Vegas: Emulation and experiment. *Computer Communications Review*, 25(4):185–195, Oct. 1995.

[3] A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan. Priority encoding transmission. In *IEEE Symposium on Foundations of Computer Science*, pages 604–612, 1994.

[4] M. Allman, V. Paxson, and W. Stevens. TCP congestion control. Request for Comments 2581, Internet Engineering Task Force, Apr. 1999.

[5] H. Balakrishnan, H. Rahul, and S. Seshan. An integrated congestion management architecture for Internet hosts. In *Proc. ACM SIGCOMM*, Cambridge, MA, USA, September 1999.

[6] J. C. R. Bennett and H. Zhang. WF2Q: worst-case fair weighted fair queueing. In *Proceedings of INFOCOM*, pages 120–128, San Fransisco, California, Mar. 1996.

[7] D. P. Bertsekas and R. G. Gallager. *Data Networks*. Prentice Hall, Englewood Cliffs, NJ, USA, 1992.

[8] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated service. Request for Comments 2475, Internet Engineering Task Force, Dec. 1998.

[9] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang. Recommendations on queue management and congestion avoidance in the internet. Request for Comments 2309, Internet Engineering Task Force, Apr. 1998.

[10] L. S. Brakmo and S. W. O'Malley. TCP Vegas: New techniques for congestion detection and avoidance. In *Proceedings of SIGCOMM*, pages 34–35, London, United Kingdom, Aug. 1994. ACM.

[11] B. Carpenter and K. Nichols. Differentiated services in the Internet. Work in progress.

[12] A. K. Choudhury and E. L. Hahne. Dynamic queue length thresholds in a shared memory ATM switch. In *Proceedings of INFOCOM*, San Fransisco, California, Mar. 1996.

[13] D. Clark and J. Wroclawski. An approach to service allocation in the internet. Technical report, IEFT, Work in Progress, August 1997.

[14] C. F. J. Corpuz. Pollution tax for controlling emissions from the manufacturing and power generation sectors: Metro manila. Technical report, School of Economics, University of the Philippines Diliman, Sept. 1999.

[15] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. *Journal of Internetworking Research and Experience*, pages 3–26, Oct. 1990. Also in Proc. ACM SIGCOMM'89, pp 3-12.

[16] S. Floyd. Congestion control principles. Request for Comments 2914, Internet Engineering Task Force, Sept. 2000.

[17] S. Floyd and K. Fall. Promoting the use of end-to-end congestion control in the internet. *IEEE/ACM Transactions on Networking*, August 1999.

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

[19] D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, October 1991.

[20] S. J. Golestani. A self-clocked fair queuing scheme for broadband applications. In *Proceedings of INFOCOM*, pages 636–646, April 1994.

[21] G. Hardin. The tragedy of the commons. *Science*, 162:1243–1248, 1968.

[22] V. Jacobson. Congestion avoidance and control. *Computer Communications Review*, 18(4):314–329, Aug. 1988. Proceedings of the Sigcomm '88 Symposium in Stanford, CA, August, 1988.

[23] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, and R. Viswanathan. ERICA switch algorithm: A complete description. Technical report, ATM Forum/96-1172, August 1996.

[24] F. Kelly. Network routing. *Philosophical Transactions of the Royal Society A337*, pages 343–367, 1991.

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

[26] Y. A. Korilis, A. A. Lazar, and A. Orda. The designer's perspective to noncooperative networks. In *Proceedings of INFOCOM*, Boston, Massachusetts, Apr. 1995.

[27] R. La and V. Anantharam. Charge-sensitive TCP and rate control in the Internet. In *Proceedings of INFOCOM*, Tel Aviv, Israel, Mar. 2000.

[28] D. Lin and R. Morris. Dynamics of random early detection. In *Proceedings of the ACM SIGCOMM'97 Conference*, pages 127–138, New York, September 1997.

[29] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. TCP selective acknowledgement options. Request for Comments 2018, Internet Engineering Task Force, Oct. 1996.

[30] A. J. Miller. *Socially Responsible Investing : How to Invest With Your Conscience*. NYIF, 1991.

[31] P. P. Mishra and H. Kanakia. On hop by hop rate-based congestion control scheme. *IEEE/ACM Transactions on Networking*, September 1996.

[32] J. Mo, R. La, V. Anantharam, and J. Walrand. Analysis and comparison of TCP Reno and Vegas. In *Proceedings of INFOCOM*, New York, Mar. 1999.

[33] J. Nagle. Congestion control in IP/TCP internetworks. Request for Comments 896, Internet Engineering Task Force, Jan. 1984.

[34] A. Orda, R. Rom, and N. Shimkin. Competitive routing in multiuser communication networks. *ACM/IEEE Transactions on Networking*, 1(5):510–521, Oct. 1993.

[35] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control - the single node case. In *Proceedings of INFOCOM*, pages 915–924. IEEE, May 1992.

[36] J. Postel. Transmission control protocol. Request for Comments 793, Internet Engineering Task Force, Sept. 1981.

[37] W. Prue and J. Postel. Something a host could do with source quench: The source quench introduced delay (SQuID). Request for Comments 1016, Internet Engineering Task Force, July 1987.

[38] K. Ramakrishnan and S. Floyd. A proposal to add explicit congestion notification (ECN) to IP. Request for Comments 2481, Internet Engineering Task Force, Jan. 1999.

[39] K. Ramakrishnan and R. Jain. A binary feedback scheme for congestion avoidance in computer networks,. *ACM Trans. on Computer Systems*, 8(2):158–181, May 1990.

[40] A. Romanow and S. Floyd. Dynamics of TCP traffic over ATM networks. *IEEE Journal on Selected Areas of Communications*, 13(4):633–641, May 1995.

[41] N. Semret, R. R.-F. Liao, A. T. Campbell, and A. A. Lazar. Peering and provisioning of differentiated Internet services. In *Proceedings of INFOCOM*, Tel Aviv, Israel, Mar. 2000.

[42] S. Shenker. Making greed work in networks: A game-theoretic analysis of switch service disciplines. In *Proceedings of SIGCOMM*, pages 47–57, London, UK, Sept. 1994.

[43] D. Stiliadis and A. Varma. Design and analysis of frame-based fair queuing: A new traffic scheduling algorithm for packet switched networks. In *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 104–115, New York, May 1996.

[44] I. Stoica, S. Shenker, and H. Zhang. Core-stateless fair queueing: Achieving approximately fair bandwidth allocations in high speed networks. In *ACM Sigcomm '98*, Septermber 1998.

[45] B. Suter, T. Lakshman, D. Stiliadis, and A. Choudhury. Buffer management schemes for supporting TCP in gigabit routers with per-flow queueing. *IEEE Journal on Selected Areas in Communications*, 17(6):1159–1169, June 1999.

[46] K. Swisher. *AOL.COM.* Times Books, September 1999.

[47] A. Tanenbaum. *Computer Networks.* Prentice Hall, 2nd Edition, 1989.

[48] T. van Vuren and M. Smart. Route guidance and road pricing - problems, practicalities and possibilities. *Transport Reviews*, 10:269–283, 1990.

[49] H. Yaiche, R. R. Mazumdar, and C. Rosenberg. A game theoretic framework for bandwidth allocation and pricing in broadband networks. *IEEE/ACM Transactions on Networking*, 8(5):667–678, October 2000.

[50] C.-Q. Yang and A. V. S. Reddy. A taxonomy for congestion control algorithms in packet switching networks. *IEEE Network Magazine*, 9(5), July/August 1995.

[51] H. Zhang and J. C. R. Bennett. Hierarchical packet fair queueing algorithms. In *Proceedings of ACM SIGCOMM*, pages 143–156. ACM, September 1997.

[52] L. Zhang. Virtual clock : A new traffic control algorithm for packet switching networks. *ACM Transactions on Computer Systems*, 9:101–124, May 1991.

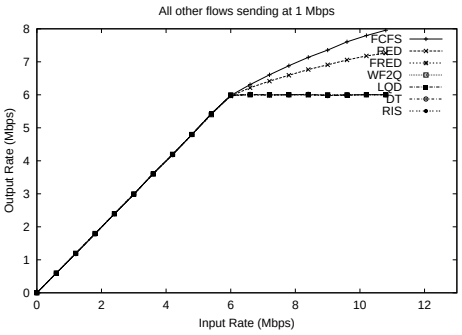

Figure 1: Uncongested link

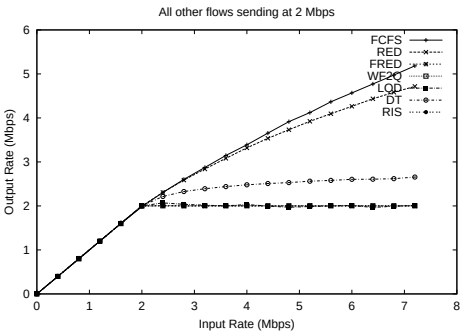

Figure 2: Critically congested link

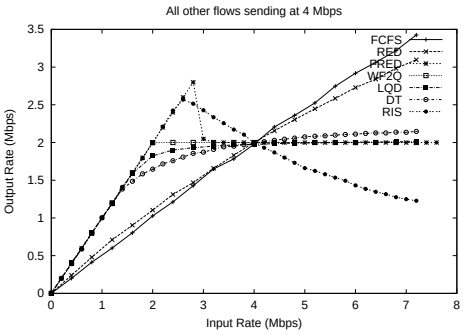

Figure 3: Congested link

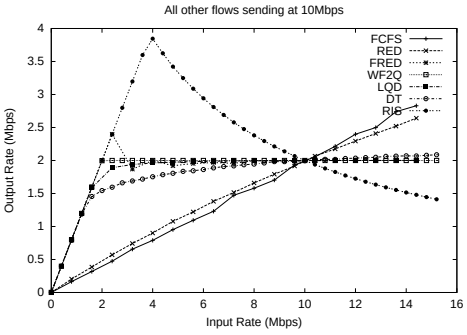

Figure 4: Heavily congested link

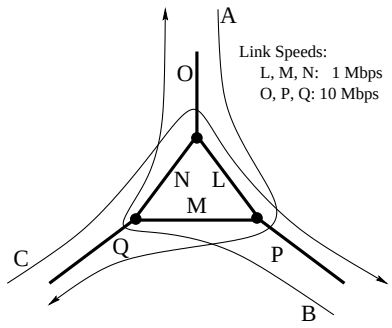

Figure 5: Congestion collapse in a sample network

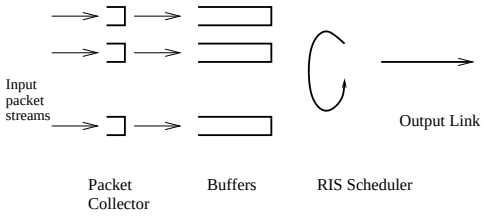

Figure 6: Hypothetical model for RIS with discrete packet boundaries

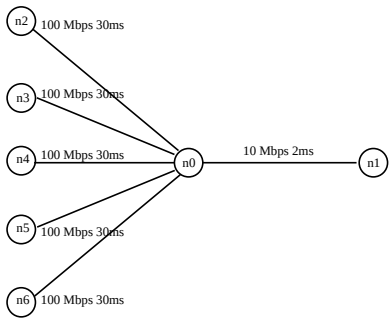

Figure 7: Simulation Scenario

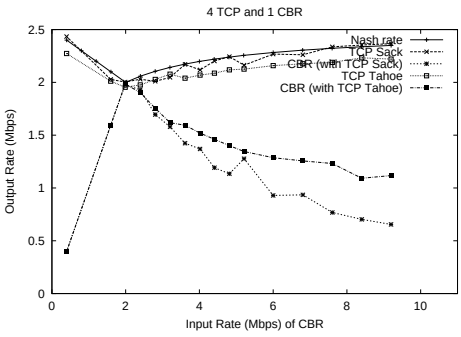

Figure 8: PRIS Performance in the presence of unresponsive flows

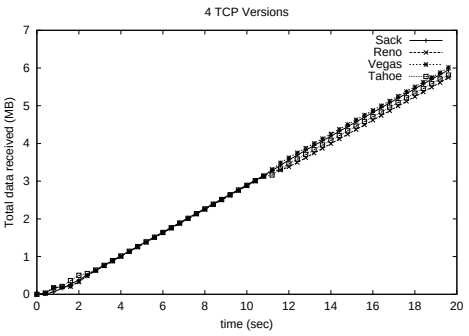

Figure 9: PRIS Performance in the presence of different TCP versions

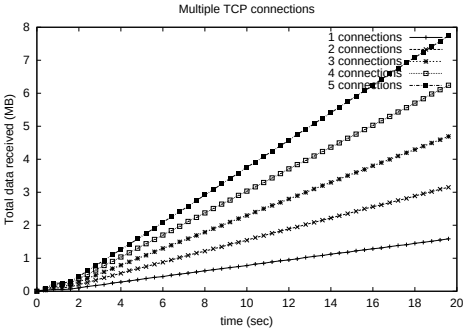

Figure 10: Drop-Tail Performance in the presence of multiple connections

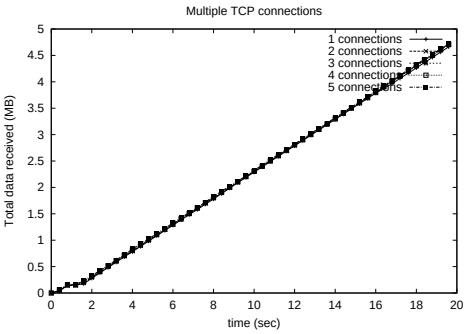

Figure 11: PRIS Performance in the presence of multiple connections

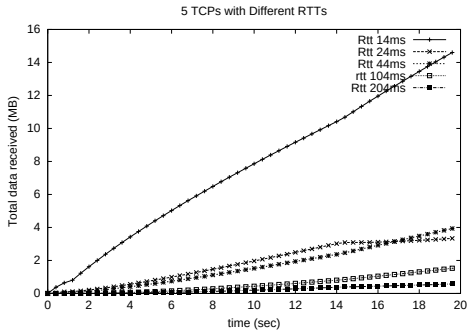

Figure 12: Drop-Tail Performance in the presence flows with varying RTTs

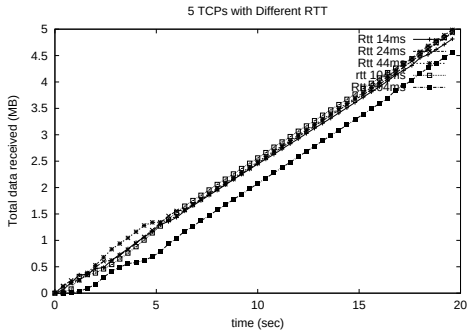

Figure 13: PRIS Performance in the presence flows with varying RTTs

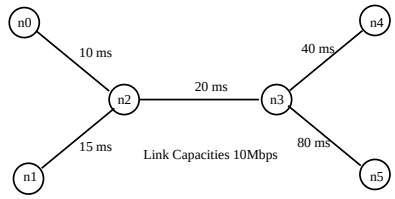

Figure 14: Simulation scenario for a network

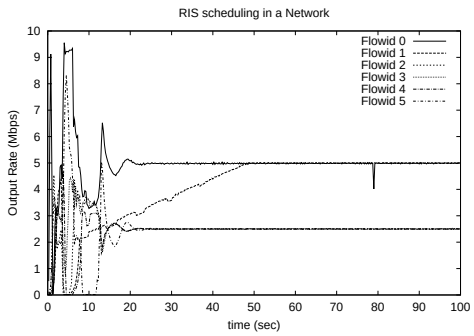

Figure 15: Output rates of flows in a network with PRIS

8 Appendix

Before proceeding to the proofs of our main theorems, we prove certain properties.

Lemma 8.1 *The congestion characteristic function $G(x, \underline{r})$, as defined in Eq. 9, is strictly increasing in $(0, r_{max})$ where $r_{max} = max(r_1, r_2, ..., r_N)$ and is constant when $x \geq r_{max}$.*

Proof: Consider the case when $0 < x < r_{max}$. Now, $\exists_i$ s.t. $r_i > x$. Now consider $x' > x$. Then $\forall_{j \neq i} min(r_j, \frac{x'^2}{r_j}) \geq min(r_j, \frac{x^2}{r_j})$ and $min(r_i, \frac{x'^2}{r_i}) > min(r_i, \frac{x^2}{r_i})$. So $G(x', \underline{r}) > G(x, \underline{r})$.

Consider the case when $x \geq r_{max}$. Now, $G(x, \underline{r}) = \sum_i r_i$. So G is constant when $x \geq r_{max}$. □

Proof (Theorem 4.1): [Uniqueness] If $\sum_{i=1}^{N} r_i \leq C$, then f is unique by definition. Using Lemma 8.1, $G(x, \underline{r})$ is an monotonically increasing function of x in the domain $[0, r_{max}]$. Now if $\sum_{i=1}^{N} r_i > C$ then $G(r_{max}, \underline{r}) = \sum_i r_i > C \Rightarrow f < r_{max}$. Therefore, f is unique.
[Output rate] If $\sum_{i=1}^{N} r_i \leq C$, from the flow conservation and the work conservation properties of RIS scheduler, $\gamma_i = r_i \leq \frac{C^2}{r_i}$. Since $f = C$, it follows that

$$\gamma_i = \min\left(\frac{f^2}{r_i}, r_i\right)$$

.

If $\sum_{i=1}^{N} r_i > C$, then there must exist at least one flow i such that $\gamma_i < r_i$. Flow i will have a non-zero backlog. From the RIS fairness property, we have $r_i \gamma_i = r^2$ where r is a constant. Therefore, $\gamma_i = r^2/r_i < r_i$. It follows that,

$$\gamma_i = \min\left(\frac{r^2}{r_i}, r_i\right). \tag{16}$$

Now consider a flow j such that $\gamma_j = r_j$. From RIS fairness we get $r_j \gamma_j \leq r^2$. Therefore, $\gamma_j = r_j \leq \frac{r^2}{r_j}$. It follows that:

$$\gamma_j = \min\left(\frac{r^2}{r_j}, r_j\right) \tag{17}$$

Since the scheduler has a positive backlog, from the flow conservation and work conservation properties of RIS, $\sum_{i=1}^{N} \gamma_i = C$. Substituting values of γ_is and r_js from Eq 16 and 17 we get:

$$\sum_{i=1}^{N} \min\left(\frac{r^2}{r_i}, r_i\right) = C.$$

Since the fair rate f is unique rate satisfying the equation $G(x, \underline{r}) = C$, we get $r = f$. Therefore, $\gamma_i = min(f^2/r_i, r_i)$ for all flows i. □

Proof (Lemma 4.4): If $\sum_{i=1}^{N} r_i \leq C$, then we have $\forall_i \gamma_i = r_i$ from the flow conservation and work conservation properties of the RIS scheduler. If r_j is decreased to r'_j, then $\sum_{i=1(i \neq j)}^{N} r_i + r'_j \leq C$ still holds. Therefore, $\gamma'_j = r'_j$ even now.

We now consider the case when $\sum_{i=1}^{N} r_i > C$. Again from the flow conservation and work conservation properties of the RIS scheduler, we have $\sum_{i=1}^{N} \gamma_i = C$. Using Eq 10 and Corollary 4.1 we get $\sum_m r_m + \sum_n f^2/r_n = C$ where for all flows m, $r_m \leq f$ and for all flows n, $r_n > f$. Therefore,

$$f^2 = \frac{C - \sum_m r_m}{\sum_n 1/r_n} \tag{18}$$

We see that the value of f also changes to f' on changing the value of r_j. We have the following cases:

$r_j \leq f$: Since $r_j \leq f$, j is one among the m flows. We rewrite Eq 18 in the form

$$f^2 = \frac{C - \sum_{m(m \neq j)} r_m - r_j}{\sum_n 1/r_n} \tag{19}$$

Hence we see that f increases on decreasing r_j. Therefore, $r'_j < r_j \leq f < f'$. Since $r'_j < f'$, using Corollary 4.1 we get $\gamma'_j = r'_j$.

$r_j > f$: Since $r_j > f$, j is one among the n flows. We rewrite Eq 18 in the form

$$f^2 = \frac{C - \sum_m r_m}{\sum_{n(n \neq j)} 1/r_n + 1/r_j} \tag{20}$$

We again have 2 cases, if $r'_j \leq f'$, then using Corollary 4.1 $\gamma'_j = r'_j$. If $r'_j > f'$, we show that $\gamma'_j > \gamma_j$. Denote the residual capacity by $C_{res} = C - \sum_m r_m$. Since $r_j > f$ and $r'_j > f'$ we have,

$$\gamma_j = f^2/r_j = \frac{C_{res}}{(r_j \sum_{n(n \neq j)} 1/r_n + 1)} \tag{21}$$

$$\gamma'_j = f'^2/r'_j = \frac{C_{res}}{(r'_j \sum_{n(n \neq j)} 1/r_n + 1)} \tag{22}$$

using Eq 20. Since $r_j > r'_j$, we get $\gamma'_j > \gamma_j$.

□

Proof (Lemma 5.1): If $r > C/N$ then $\eta(r)$ satisfies:

$$\frac{x^2}{r} + (N-1)x = C \tag{23}$$

It is easy to see that there is exactly one positive real root of the above equation.

1. Suppose $\eta(r) \geq r$. Then LHS of equation 23 becomes $\eta^2(r)/r + (N-1)\eta(r) \geq Nr > C$. So equation 23 can never be satisfied. Hence $\eta(r) < r$.

2. From equation 23, it is clear that when $r > C/N$, $\eta(r) > C/N$. Now $G(\eta(r), \underline{r}) = \sum_i min(r_i, \frac{\eta^2(r)}{r_i}) \leq \frac{\eta^2(r)}{r} + (N-1)\eta(r) = C$. It follows that $f \geq \eta(r)$ since $G(x, \underline{r})$ is an increasing function of x.

□

Proof (Theorem 5.1): [Nash Equilibrium] Let $\forall_i r_i = C/N$. Then $\forall_i \gamma_i = C/N$ since $f \geq C/N$ always. WLOG, suppose that user 1 changes its input rate to r'_1.

Case 1: $r'_1 < C/N$**:** Clearly, $\gamma_1 < C/N$.

Case 2: $r'_1 > C/N$**:** Now $\sum_{i=1}^{N} r_i > C$, so the scheduler will have a positive backlog. Now $\gamma_{i \in (2..N)} = min(C/N, \frac{f^2}{C/N}) = C/N$ since $f \geq C/N$. So, from the flow conservation and work conservation properties $\gamma_1 = C - \sum_{i \in (2..N)} \gamma_i = C/N$.

So user 1 can never increase its output rate by unilaterally changing its input rate. Hence $\forall_i r_i = C/N$ is a Nash equilibrium.

[Uniqueness] Consider a vector of input rates $\underline{r}$ constuting a Nash Equilibrium, s.t $\exists_i r_i \neq C/N$. There are two cases:

Case 1: $\exists_i r_i < C/N$**:** Here $\gamma_i < C/N$. So by increasing r_i to C/N, γ_i can be increased to C/N. Therefore $\underline{r}$ does not constitute a Nash Equilibrium.

Case 2: $\forall_j r_j \geq C/N$ **and** $\exists_{\mathbf{i}} \mathbf{r_i} > \mathbf{C/N}$**:** WLOG, assume that $r_1 = max(r_1, r_2, ...r_N) > C/N$.

1. $\exists_{\mathbf{i \geq 2}} \gamma_\mathbf{i} < \eta(\mathbf{r_1})$: Using Lemma 5.1, $\forall_{r_2, r_3 ... r_N} f \geq \eta(r_1)$. So by making $r_i = \eta(r_1)$, γ_i can be increased to $\eta(r_1)$.
2. $\forall_{\mathbf{i \geq 2}} \gamma_\mathbf{i} = \eta(\mathbf{r_1})$: $\gamma_1 = C - \sum_{i \in (2,N)} \gamma_i = \frac{\eta^2(r_1)}{r_1} = C - (N-1)\eta(r_1) < C/N$. So by making $r_1 = C/N$, γ_1 can be increased to C/N.
3. $\forall_{\mathbf{i \geq 2}} \gamma_\mathbf{i} \geq \eta(\mathbf{r_1})$ and $\exists_{\mathbf{j \geq 2}} \gamma_\mathbf{j} > \eta(\mathbf{r_1})$: It is easy to see that $f < r_{max} = r_1$. From Lemma 5.1 $f \geq \eta(r_1)$. $\Rightarrow \gamma_1 = \frac{f^2}{r_1} \geq \frac{\eta^2(r_1)}{r_1}$. Now $\sum_i \gamma_i = \gamma_1 + \gamma_j + \sum_{i \geq 2, i \neq j} \gamma_i > \eta^2(r_1)/r_1 + \eta(r_1) + (N-2)\eta(r_1) = C$. So this case is not possible.

It follows that $\forall_i r_i = C/N$ is the unique Nash equilibrium. □

Proof (Theorem 5.3):

1. $\mathbf{r_1 \leq C/N}$: In this case $\gamma_1 = r_1$, irrespective of the value of r_1 because $f \geq C/N$. The spare capacity $C_{res} = C - r_1$, will be divided among the remaining users. Using Theorem 5.1, the unique Nash equilibrium for the remaining users is when each sends at $\frac{C_{res}}{N-1}$ which is $\eta(r_1)$.
2. $\mathbf{r_1 > C/N}$: Let $\forall_{i \in (2,N)} r_i = \eta(r_1)$. We now show that this vector of input rates constitute a Nash Equilibrium. Using Lemma 5.1, $f \geq \eta(r_1)$ so $\gamma_{i \in (2,N)} = \eta(r_1)$. WLOG, assume that user 2 deviates its rate from $r_2 = \eta(r_1)$ to r_2'.
   (a) $\mathbf{r_2' < \eta(r_1)}$: Here $\gamma_2 < \eta(r_1)$ and by making $r_2 = \eta(r_1)$, user 2 can get output rate $\eta(r_1)$.
   (b) $\mathbf{r_2' > \eta(r_1)}$: If $\underline{r}'$ is the new input rate vector, then $G(\eta(r_1), \underline{r}') = \frac{\eta^2(r_1)}{r_1} + \frac{\eta^2(r_1)}{r_2'} + (N-2)\eta(r_1) < \frac{\eta^2(r_1)}{r_1} + (N-1)\eta(r_1) = C \Rightarrow f > \eta(r_1)$. Now $\gamma_1 = \min(r_1, \frac{f^2}{r_1}) > \frac{\eta^2(r_1)}{r_1}$ and so $\gamma_2 = C - \sum_{i \neq 2} \gamma_i < C - (\frac{\eta^2(r_1)}{r_1} + (N-2)\eta(r_1)) = \eta(r_1)$. So the output rate of user 2 always decreases when it deviates from $\underline{r}$. Hence $\underline{r}$ is a Nash equilibrium for the N - 1 user subsystem.

   We now show that the above Nash Equilibrium is unique.

   (a) $\sum_{\mathbf{i=1}}^{\mathbf{N}} \gamma_\mathbf{i} < \mathbf{C}$: Clearly $\exists_j$ s.t. $\gamma_j = r_j < C/N$. Now, r_j can be increased to C/N to get $\gamma_j = C/N$.
   (b) $\sum_{\mathbf{i=1}}^{\mathbf{N}} \gamma_\mathbf{i} \geq \mathbf{C}$ **and** $\exists_{\mathbf{i \in (2,N)}} \mathbf{r_i} \neq \eta(\mathbf{r_1})$: Using a similar argument as above, we get $f > \eta(r_1)$. Now, $\gamma_1 = f^2/r_1 > \eta^2(r_1)/r_1$. From the flow conservation and work conservation properties, $\sum_{i \in (2,N)} \gamma_i = C - \gamma_1 < C - \frac{\eta^2(r_1)}{r_1} = (N-1)\eta(r_1)$.
   $\Rightarrow \exists_j \gamma_j < \eta(r_1)$. Since by making $r_j = \eta(r_1)$, γ_j can be made equal to $\eta(r_1)$, this can never be a Nash equilibrium for the N-1 user subsystem.

Hence if user 1 sends at r_1, then $\forall_{i \in (2,N)} r_i = \eta(r_1)$ is the unique Nash equilibrium for the remaining N-1 users. □

Proof (Theorem 5.2): WLOG assume that user 1 becomes the leader and sends at r_1 in a Stackelberg Equilibrium.

1. $r_1 < C/N$: $\gamma_1 < C/N$. This cannot be the case since $f \geq C/N$ and making $r_1' = C/N$ makes $\gamma_1' = C/N > \gamma_1$.

2. $r_1 > C/N$: Using Theorem 5.3, the unique Nash Equilibrium for the remaining flows is when each sends at $\eta(r_1)$. From Lemma 5.1 $\eta(r_1) > C/N \Rightarrow \gamma_1 \leq C - \sum_{i\in(2,N)} \gamma_i = C-(N-1)\eta(r_1) < C-(N-1)C/N = C/N$. This case is impossible as making $r_1' = C/N$ makes $\gamma_1' = C/N > \gamma_1$. The only case left is the following.

3. $r_1 = C/N$: $\gamma_1 = C/N$ since $f \geq C/N$ always.

So every Stackelberg Equilibrium will have $r_1 = C/N$. Since the unique Nash Equilibrium for other flows is $\forall_{i\in(2,N)} r_i = \eta(r_1) = C/N$, it follows that the only Stackelberg equilibrium is when $\forall_i r_i = C/N$. □

Lemma 8.2 *If $r_j = \gamma_j$, then removal of flow j and adjusting link capacity as $C' = C - r_j$ does not change the fair rate.*

Proof: From Eq. 10, the fair rate f satisfies $G(f, \underline{r})$. This can be rewritten as $\sum_{i\neq j} min(r_i, \frac{f^2}{r_i}) = C - \gamma_j = C - r_j$. Substituting the new cpacity $C' = C - \gamma_j$ and the new rate vector $\underline{r}'$ we get $G(f, \underline{r}') = C'$. Since f is the fair rate satisfying $G(f, \underline{r}') = C'$ the fair rate remains unchanged. □

Before proving Lemma 5.2 we give a brief overview of max-min fair rate computation in an arbitrary network. Let C_ℓ denote the capcacity of link ℓ and N_ℓ denote the number of flows passing through link ℓ. Let F_ℓ be the set of flows passing through link ℓ.

The algorithm to compute max-min fair rates [7] works as follows. At every stage k (k is initially 1), it finds the minimum bottleneck link ℓ s.t. for all other links ℓ', $C_\ell^k/N_\ell^k \leq C_{\ell'}^k/N_{\ell'}^k$. WLOG, it labels this link as ℓ_k. For all residual flows P^k passing through ℓ_k (k = 1), it assigns $\mathcal{M}_{i\in P^k} = C_\ell^k/N_\ell^k$, where C_ℓ^k and N_ℓ^k are respectively the residual capacity and the residual number of flows passing through the link ℓ_k. The link ℓ_k and all these flows are removed and the residual capacities and residual number of flows of all the other links are updated as follows:

$$C_\ell^{k+1} = C_\ell^k - \sum_{i\in P^k, i\in F_\ell} \mathcal{M}_i \tag{24}$$

$$N_\ell^{k+1} = N_\ell^k - \sum_{i\in P^k, i\in F_\ell} 1 \tag{25}$$

Initially, $C_\ell^1 = C_\ell$ and $N_\ell^1 = N_\ell$. The algorithm terminates when there are no more flows to be removed.

Proof (Lemma 5.2): Consider the set of flows F_{ℓ_1} through the minimum botleneck link ℓ_1.

Claim 1 *In any Nash or Stackelberg Equilibrium without losses, flows $i \in F_{\ell_1}$ will have $r_i = \gamma_i = C_{\ell_1}/N_{\ell_1}$.*

Proof: Since there are no losses, $\forall_i \gamma_i = r_i$ and $\sum_{i\in F_{\ell_1}} r_i \leq C_{\ell_1}$.

Assume for contradiction, $\exists i \in F_{\ell_1}$ s.t. $r_i < C_{\ell_1}/N_{\ell_1}$.

Since ℓ_1 is the minimum bottleneck link, the fair rate f_ℓ of any other link ℓ as defined by Eq. 10 is greater than or equal to C_{ℓ_1}/N_{ℓ_1}. Set r_i' equal to C_{ℓ_1}/N_{ℓ_1} and the new output rate at each link will satisfy $\gamma_i' = r_i' > \gamma_i$. So this profile of rates cannot constitute a Nash equilibrium.

Now assume WLOG, that flow $1 \in F_{\ell_1}$ becomes the leader in a Stackelberg Equilibrium. If $r_1 = C_{\ell_1}/N_{\ell_1}$, then using a similar argument as above, Nash Equilibrium of the remaining flows $i \in F_{\ell_1}, i \neq 1$ is when $r_i = C_{\ell_1}/N_{\ell_1}$. In this case γ_1 becomes equal to C_{ℓ_1}/N_{ℓ_1}. So any Stackelberg Equilibrium with flow 1 as leader should have $\gamma_1 \geq C_{\ell_1}/N_{\ell_1}$.

If $r_1 > C_{\ell_1}/N_{\ell_1}$ then the Nash rate of the remaining flows will be atleast C_{ℓ_1}/N_{ℓ_1}. Therefore flow 1 will have losses. Hence the proof. □

To extend Claim 1 to the subsequent bottleneck links, we use the flow removal property of Lemma 8.2. Since the flows in F_{ℓ_1} do not get any losses, they may be removed from the network without changing the fair rate f at any other link.

Therefore we have the following property:

In any Nash or Stackelberg Equilibrium without losses, any flow $i \in F_{\ell_k}$ s.t. $i \notin F_{\ell_{k'}}$ where $k' < k$ will have $\gamma_i = r_i = C^k_{\ell_k}/N^k_{\ell_k} = \mathcal{M}_i$. This is the set of max min fair rates and will hence constitute the unique Nash and Stackelberg Equilibrium without losses. □

Proof (Theorem 5.4): Follows directly from Lemma 5.2. □

Lemma 8.3 *Let U_i be the utility function for user i. If $\forall_i U_i \in U^{\gamma l}$, then in any Nash or Stackelberg equilibrium, no user will experience losses.*

Proof:

- **Nash**: Suppose that for user i, $\gamma_i < r_i$. So the loss-rate is $l_i = r_i - \gamma_i > 0$. If user i changes its input rate to $r'_i = \gamma_i$, its input rate at the first link is decreased. Using Lemma 4.4 if the output rate should either stay the same, or increase, or become equal to the new input rate. Since $r'_i = \gamma_i$, the output rate at this link becomes equal to its new input rate, γ_i. The same argument applies to all the following links in the path. Therefore $\gamma'_i = \gamma_i$. The new loss rate is therefore zero. Since loss rate is decreased and output rate is same as before, user i's utility increases. So in any Nash equilibrium, no user can have losses.
- **Stackelberg**: In any Stackelberg equilibrium, the leader cannot have any losses, else it can increase its utility by decreasing the input rate to its current output rate. All the other flows are in Nash equilibrium, so by a similar argument they also cannot have any losses.

□

Proof (Theorem 5.5): From Lemma 8.3, any Nash or Stackelberg equilibrium with utility functions $U_i \in U^{\gamma l}$ cannot have losses. If the loss rate is zero, then the a user's utility becomes only a function of its output rate and hence belongs to U^{γ}. Then using Lemma 5.2, it follows that the unique Nash and Stackelberg equilibrium is at $\underline{\mathcal{M}}$. □