An Incrementally Deployable Approach for Achieving Fair Rate Allocations

Yannick Blanpain Vaidyanathan Anantharaman Raghupathy Sivakumar

School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA 30332, USA
gte463w@prism.gatech.edu, vaidy@ece.gatech.edu, siva@ece.gatech.edu
http://www.ece.gatech.edu/research/GNAN

Abstract—**Several router mechanisms have been proposed to achieve fair rate allocations. However, most of them require the proposed mechanisms to be deployed at all routers within an autonomous domain in order to achieve the desired, or any appreciably improved, quality of service. In this paper, we present an incrementally deployable QoS architecture called *iQ*. iQ consists of a set of core- and edge-router mechanisms that allow for changing one router at a time, improving the rate fairness provided by the network gracefully, for each additional iQ router deployed. We use simulations to compare the incremental deployability of the iQ architecture with that of existing approaches.**

I. Introduction

Several promising approaches for providing fair rate allocations have been proposed in literature. These include core-stateless approaches like CSFQ [1], and *Corelite* [2] that do away with any per-flow processing and still achieve approximately fair rate allocations.. While the core-stateless approaches[1] are clearly desirable by virtue of their ability to scale with increasing line-speeds and number of flows in the network, they require the routers to be upgraded "one island at a time". In other words, such schemes have typically assumed the ability to change all routers within an autonomous domain simultaneously. Furthermore, while the per-flow processing approaches do not explicitly assume the changing of all routers simultaneously within an autonomous domain, to the best of our knowledge no work has been done to evaluate their performance under partially deployed scenarios. In this paper, we approach the problem of fair rate allocation from the perspective of *incremental deployability*. We define incremental deployability as the ability of a rate allocation scheme to gracefully improve the fair service it provides to flows, when being deployed *one router at a time*.

Although the motivation for providing such incremental deployability can arise from numerous factors including economic considerations, limited requirements of peer-networks, or merely logistic constraints involved in the upgrade of a large-scale backbone network, the absolute necessity of incremental deployability is arguable. However, our focus in this paper is not to motivate incremental deployment, but to look at the implications on the design of rate allocation schemes if incremental deployability were to be a necessity, or even a desirable characteristic.

The contributions of this paper are twofold:

1. We characterize the incremental deployability of schemes belonging to the two categories mentioned earlier, and conclude the following: (a) Although per-flow processing schemes exhibit good incremental deployability, their performance in this regard can be improved upon with appropriate enhancements. However, notwithstanding their incremental deployability, such schemes are less likely to be deployed in backbone networks due to their high per-flow processing overheads. (b) Current core-stateless approaches do not exhibit good incremental deployability, inspite of having other desirable properties, including scalability and low-complexity.
2. We present an incrementally deployable fair rate allocation approach called *iQ*. iQ consists of a set of core-router and edge-router mechanisms that in tandem, allow for the "fair rate allocation service" to be incrementally deployed, one router at a time. In particular, iQ allows for the co-existence of legacy[2] routers along with iQ routers in an autonomous domain, and supports the co-existence of legacy flows[3] along-with flows that have subscribed to the fair-rate service. Finally, iQ achieves the incremental deployability while still adhering to the "stateless-core principle". We show through simulation results obtained over real backbone network topologies that iQ exhibits good incremental deployability, providing graceful improvement in the network's fairness properties with each upgraded router.

The rest of the paper is organized as follows: In Section II, we discuss in more detail the notion of incremental deployability, and provide some insights into the reasons for current schemes not being incrementally deployable. In Section III we describe the iQ approach, including the core-router and edge-router mechanisms. In Section IV, we present simulation results comparing the performance of iQ with that of existing schemes. In Section V, we

[1] We borrow the term *core-stateless* from [1] and use it to classify the set of approaches that do not perform per-flow processing in the core.

[2] We assume drop-tail as the default mechanism for rate allocation.

[3] Possibly from peer-networks that do not need any rate quality of service.

conclude the paper.

II. Incremental Deployability and Goals

In this section we discuss some of the problems that need to be addressed by an incrementally deployable rate allocation scheme. We then identify certain goals that need to be achieved in order for the allocation scheme to be able to solve the problems. In the rest of the paper, we distinguish between two types of routers in an autonomous domain: (i)*Edge routers* that interface with peer-networks, and act as ingress and egress points for flows that transit through the autonomous domain; and (ii) *Core routers* that serve to switch traffic between edge routers and other core routers within the autonomous domain.

We define incremental deployability as the ability of a rate allocation scheme to provide increasingly better fair service with each additional router upgraded[4] in an autonomous domain. We now present some of the problems that need to be addressed in order for an allocation scheme to be able to provide such service. We refer to routers that have been upgraded as *fair-routers*, and the non-upgraded routers as *legacy-routers*. Similarly, we refer to flows requiring fair service as *fair-flows*, and the default flows as *legacy-flows*.

A. Impact of Legacy Core Routers

In a partially upgraded network, fair-routers will have to co-exist with other legacy-routers. The impact of the presence of legacy-routers on the performance of fair-routers depends upon the specific allocation scheme employed at the fair-routers. In core-stateless approaches, fair allocation of link bandwidth at core routers is achieved by making use of *dynamic state information* about the flows. Edge routers pass the dynamic state to core routers through varying schemes including fields in the packet headers (dynamic packet state) [1], or specialized packets in the flow (dynamic flow state) [2]. The core routers rely solely on the dynamic state carried by the flow, and hence do not perform any per-flow processing. However, in the presence of legacy-routers, such approaches encounter the following problems that adversely affect the fair service they offer:

- Legacy-routers drop packets indiscriminately[5]. Hence, flows traversing such routers will inherently receive unfair service. While this by itself cannot be completely overcome[6], it leads to unfair service even at fair routers. We elaborate upon this phenomenon next.
- While legacy-routers drop packets indiscriminately, being unaware of the fair allocation scheme, such routers will, in addition, fail to update the dynamic state of flows when dropping packets. Hence, the dynamic state at downstream fair routers can be inconsistent with the actual flow state. When fair routers use such inconsistent dynamic state to perform rate allocation, the allocation will be unfair.
- While the unfair allocation because of inconsistent state can be plausibly perceived as a transient phenomenon (if edge routers are assumed to adapt to the fair share feedback they receive, it can be shown that flows will eventually receive their fair allocations at the fair routers), this is true only if the unfair allocations at the drop-tail routers remain stable. In other words, given an arbitrarily indiscriminate rate allocation at legacy-routers, that fluctuates with time, the fair share computation at fair routers will fail to converge, causing unfair allocation at fair routers to become a persistent phenomenon.

In approaches that maintain per-flow state, the fair rate allocation is done based on the actual incoming traffic at the concerned core router. Hence, even in the presence of upstream unfairness, a fair router will act fairly, and in the process dampen the extent of unfairness experienced by the incoming flows upstream. Thus, in such approaches, the unfairness is restricted to only at the legacy-routers. However, this desirable property is achieved at the expense of having to perform per-flow processing.

B. Impact of Legacy Edge Routers

While core routers are responsible for fair allocation, edge routers in core-stateless approaches are responsible for conveying to the core routers, the dynamic state used in their rate allocation schemes. In the event that an edge router is a legacy router, it will fail to convey any such dynamic state information to the core. Hence, the presence of such edge routers will result in *legacy-flows* co-existing with *fair-flows* in the core network. While this does not have any impact on legacy-core-routers (where no fair allocation schemes exist anyway), it obviously has a severe impact on the fairness achieved at fair-routers. Specifically, given that the legacy-flows carry no dynamic state, how should the fair routers treat the legacy-flows? Aggregating all legacy-flows into one logical flow might result in unfair allocations to legacy flows. A traffic engineering solution (wherein, the capacity of the network is partitioned between legacy- and fair-flows) might be possible, but would not be desirable in a pay-per-use service model [3].

The challenge then is to determine dynamically how the capacity at a fair router should be divided between legacy and fair-flows. Once the split is determined, it is sufficient to then provide fairness only among the fair-flows (within

[4] To implement the concerned rate allocation scheme.

[5] Recall that we have assumed the use of the drop-tail mechanism at legacy-routers.

[6] In iQ, edge routers employ a compensation model that addresses this problem.

their allocation), as the legacy-flows do not expect any fair allocation in the first place. However, it is critical for legacy flows not to be penalized in any way due to the upgrade of a part of the network[7].

In approaches that perform per-flow processing, this problem can be overcome, as flows can be distinguished using the source and destination addresses in the headers. However, this is again achieved at the cost of maintaining per-flow state and performing per-flow processing.

C. Goals

Based on our earlier discussions, we now identify the key goals that have to be achieved by an incrementally deployable fair-rate allocation scheme: (i) Fair-service that is independent of the unpredictable allocations at upstream legacy-routers; (ii) Accommodation of both legacy- and fair-flows. Service that is neither unfair to the fair-flows, nor to the aggregate of legacy-flows. Although the prevention of unfair allocation is critical as far as the aggregates are concerned, once the allocation is made, it will be the responsibility of the allocation scheme to ensure micro-level fairness only among the fair-flows; (iii) While the unfair allocation at legacy-core-routers cannot be overcome in isolation, the overall architecture should ideally consist of schemes that reduce the degree of unfairness; (iv) No maintenance of per-flow state in the core routers; (v) The performance of the allocation scheme should be at-least as good as (if not better) that of the existing core-stateless approaches in a fully deployed scenario; and (vi) The control overheads incurred in the process of achieving the incremental deployability should be limited, and bounded.

In the next section, we present the *iQ* approach that achieves the above goals in supporting fair rate allocation, and thus is incrementally deployable.

III. The iQ Architecture

A. Overview

When fully deployed, iQ approximately achieves the same fairness model that most other rate allocation approaches do - max-min fairness [2], [1]. However, the fairness model supported by iQ in partially deployed scenarios is unique to itself. Briefly, in a partially upgraded iQ network, legacy-flows observe no change in service, and continue to enjoy the rate service they received before the network was upgraded. Similarly, at every fair-core-router, the aggregate rate enjoyed by the fair-flows traversing that core-router remains the same. However, within the aggregate allocation, the core-router arbitrates the rate fairly among the fair-flows. For example, if 1 legacy-flow and 2 fair-flows at a core-router would have received a rate allocation of 30, 10, and 40 respectively if the router were a legacy-router, in the iQ fairness model, they would receive a rate allocation of 30, 25, and 25 respectively. In addition, iQ uses a compensation scheme at ingress-routers to reduce the unfairness between flows sharing the same ingress-to-egress path. Details of the compensation model will be presented later in the section. We now look at an overview of the core- and edge-router mechanisms in iQ that help in achieving this fairness model.

A.1 Core Router Mechanisms

iQ consists of a set of core router and edge-router mechanisms that in combination achieve the goals stated in the previous section. The core-router mechanism in iQ has two key components: (i) a *measurement based fairness allocation scheme* for providing fairness among fair-flows, and (ii) a *drop-tail emulation module* for determining the allocation between the legacy- and fair-flows.

- The measurement based allocation scheme is motivated by the goal to provide fair service based on the actual traffic at the core router, as opposed to the dynamic state carried by packets in the flow. In this regard, the allocation scheme is similar to per-flow schemes such as fair queuing and its variants. However, in adherence to the core-stateless principle, the core router in iQ does not maintain per-flow state or perform per-flow processing. Instead, the allocation scheme performs processing and state maintenance on a per-edge-router basis.
  Packets are labeled at the ingress edge router[8] with the router's identifier. Core routers monitor the arrival rate per ingress router, and provide fair rate allocation among flow aggregates belonging to different ingress routers. Each core router thus maintains state (a few bytes) to keep track of the arrival rate corresponding to an ingress router during an epoch. The fair rate allocation per ingress router is computed at the end of each epoch, and depends solely on the actual arrival rates during the just completed epoch, and the weights of the ingress routers. The weight of an ingress router at a core router is set proportional to the number of flows traversing the core router, that belong to the ingress router. A robust and low-overhead weight distribution scheme is used by ingress routers to periodically update core routers with the updated weights. All packets in excess of the computed fair share are dropped during the ensuing epoch. A packet accepted is labeled with the minimum of the fair share at the core router and the fair share already labeled on the packet. The core router thus provides coarse level fairness between aggregates of flows corresponding to each ingress edge router. However, the coarse level fairness in tandem

[7] Note that we perceive "unfair allocation" and "indiscriminate allocation" differently, with the former being clearly more undesirable than the latter.

[8] Here-on referred to as the ingress router.

with the per-flow arbitration scheme at edge routers (which we describe later in the section) provide approximately fair allocations for individual flows. We will describe both the weight distribution and the arbitration schemes when we describe the edge router mechanisms.

It is possible to believe based on the discussions thus far that the core router mechanisms are tantamount to maintaining significant amount of state or performing intensive processing. However, such a plausible conjecture is invalid for the following reasons: (a) Core routers in iQ do not maintain individual queues for each ingress router. Instead, they maintain a single FIFO queue which is shared by all incoming flows. (b) No classification is required upon packet arrival since iQ core routers make use of predetermined ingress-router identifiers that packets carry. Packets are labeled with these identifiers at the ingress router. (c) The state that is maintained per ingress-router is minimal. We show later in the section, that the amount of state maintained per ingress-router is just a few bytes (approximately four bytes). Given that the number of edge routers is a constant for a network topology, both the amount of state maintained and the number of identifiers required is independent of the number of flows in the network.

The measurement based fair rate allocation described thus far does not rely on dynamic packet or flow state, and hence escapes from the vagaries of the allocation characteristics of upstream legacy-routers.

- In addition to the fair rate allocation module, the core router mechanisms include a *drop-tail emulation module* whose key role is the division of bandwidth between legacy- and fair-flows. The role of the drop-tail emulation is to keep track, at any instant, of the state of the output-buffer if the router had been a legacy (drop-tail) router. The emulation module is notified of all packet arrivals and transmissions. When a packet arrives, it is first passed through the emulation module. The emulation module, based on its current drop-tail state, indicates whether the packet would have been "accepted" or "rejected". The packet is then passed to the fair rate allocation module. If the packet belongs to a legacy-flow, the fair rate allocation module honors the decision (to accept or reject) made by the emulation module. However, if the packet belongs to a fair-flow, the packet is either accepts or rejected based on the fair service mechanism described earlier. In addition, the allocation module keeps track of the rate allocation made by the drop-tail emulation module for the fair-flows. At the end of each epoch, the per-ingress fair share allocation is computed based not on the actual capacity of the output link, but based on an exponentially averaged value of the rate allocation made by the drop-tail emulation module for fair-flows during the past epochs.

  The significance of the drop-tail emulation module is that, the division of bandwidth between legacy- and fair-flows is done based on how the split would have occurred in a legacy router. While this does not prevent indiscriminate allocations for the two flow aggregates, it ensures that legacy-flows continue to enjoy exactly the same service they had received before the network upgrade. As the number of legacy-flows at a core router decreases, the aggregate allocation for fair-flows increases[9]. At an extreme, when there are no more legacy-flows, the drop-tail emulation module ceases to have an impact on the performance of the fair-rate allocation module.

A.2 Edge Router Mechanisms

Rates of individual flows, and the bottleneck fair share along a path are monitored at the egress edge router, and periodically conveyed back to the ingress router. The ingress router uses this information to perform two functions: *arbitration*, and *compensation*. First, the ingress router determines the rates of individual flows based on feedback information received. In addition, if two flows sharing the same path enjoy different individual rates (either because of the coarse level fairness provided by core routers, or because of unfair allocation at legacy core routers), the ingress router uses a compensation model to re-allocate bandwidth from the flow with the higher aggregate rate to that with the lower aggregate rate. Finally, the ingress router periodically sends a stream of special packets (possibly piggybacked on data) that carry the weight information for each output-link on a core router. The stream is activated upon the entry of new flows (that alters the weights), and is terminated upon acknowledgment of receipt of the new weights, from the egress router. Finally, the weights are computed at the ingress router based on the knowledge of the intra-domain paths that the individual flows are assigned to. In an autonomous domain that uses layer two switching, or MPLS with short cut routing, this information is already available at the edges. In the absence of such readily available information, a dedicated protocol can be used to determine routes between ingress-egress pairs[10]. Due to lack of space, we do not present the details of the iQ core and edge router algorithms. For the algorithmic details, the reader can refer to [5].

[9] Given a sufficiently large number of flows.

[10] The dedicated protocol will not incur high overhead since it is typically recommended that route flutters should be avoided within autonomous domains, and hence the protocol would be required to run only on infrequent route flutters [4].

IV. Simulation Results

In this section, we compare the performance of the iQ architecture with that of FQ and CSFQ, in terms of its incremental deployability. The *ns-2* network simulator was used for all the simulations. The iQ core router mechanisms were implemented as a queue agent in *ns*. Unless otherwise specified, all simulations were run for a period of 100 seconds. A constant packet size of 1 Kilobyte was used. The epoch size in iQ was set to one second, and the epoch size in CSFQ was set to 1 tenth of a second. We show the results for two topologies in this paper. Although simulations were performed with several other topologies and traffic scenarios, we do not present them here due to lack of space. Interested readers can refer to [5].

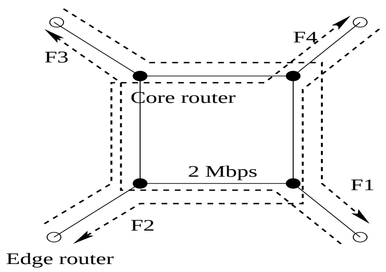

Fig. 1. 4-Node Toy Topology

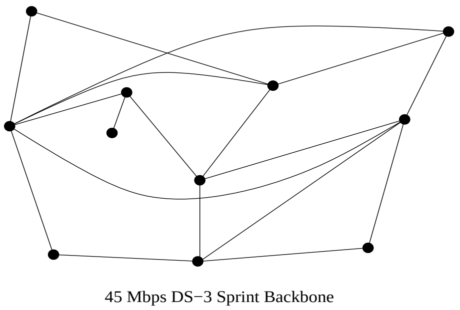

Fig. 2. Sprint Backbone Topology

We use the toy topology shown in Figure 1 and a sprint backbone topology shown in Figure 2 as our test topologies [6]. Although the original sprint network is a 45 Mbps DS-3 backbone, we use scaled down bandwidths of 10 Mbps per link (for reducing the time complexity of the simulations) Each core-router in the Sprint topology is connected to exactly one edge-router. Simulations with the Sprint topology were done with 1100 flows. Each edge-router serves as an ingress for 100 flows. The 100 flows at an ingress-router is divided into 10 sets of equal number of flows, each set having a unique egress-router among the other 10 edge routers. The rate of each flow is set to 200 kbps. Unless otherwise mentioned UDP sources were used for all traffic.

We present two sets of results in the rest of the section:

- *Core-router Incremental Deployability.* We show the evolution of the network fairness (using Jain's fairness index [7]) with increasing number of fair-core-routers in each of the three cases. Both topologies are used for these results, and all edge-routers are assumed to be fair-routers.
- *Edge-router Incremental Deployability.* We use Jain's fairness index again to demonstrate the fairness properties of the three approaches vis-a-vis the incremental deployment of fair routers at the edges. All core-routers are assumed to be fair-routers for this set of results.

A. Core-Router Incremental Deployability

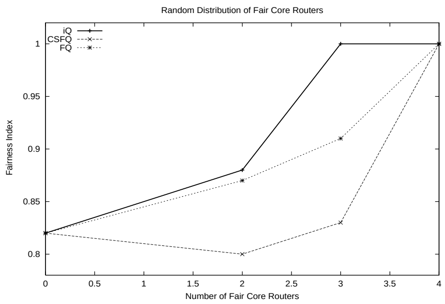

Fig. 3. Core Incremental Deployability: 4 Node Topology

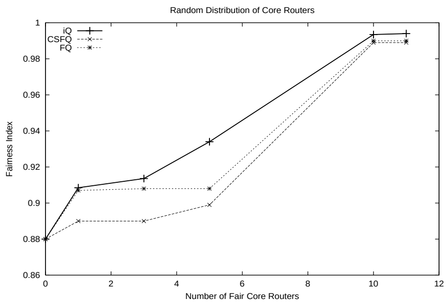

Fig. 4. Core Incremental Deployability: Sprint Topology

In Figure 3, we show the progression of the fairness index for each of the three approaches with increasing number of fair-routers in the network shown in Figure 1. Four flows are used with rates of 5 Mbps each respectively. The link bandwidths are set to 2 Mbps each. As seen, iQ performs better than CSFQ in terms of the progression. It is also interesting to note that iQ does better than FQ for some data points. The reason behind this behavior is the compensation model used in iQ. Even in the event of unfair rate allocation at legacy core-routers, the iQ compensation model ensures that flows that suffer from indiscriminate losses are compensated later. In Figure 4, we show the fairness index of the three approaches in the Sprint backbone network topology. It can again be seen that the iQ approach does better than CSFQ and FQ. The choice of the specific routers to be upgraded was done using random distribution.

B. Edge-router Incremental Deployability

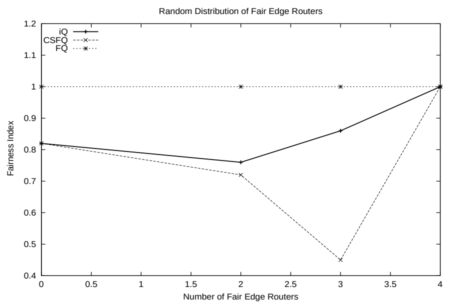

Fig. 5. Edge Incremental Deployability: 4 Node Topology

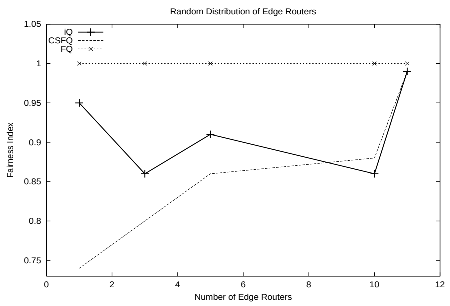

Fig. 6. Edge Incremental Deployability: Sprint Topology

In this section, we present the incremental deployability of the three approaches, at the edges. All core-routers are set to be the corresponding fair routers. The edge-routers are upgraded one at a time, and the fairness index measured. Although the same fairness index is used, we use only the fair-flows to compute the fairness index shown. The performance of the FQ approach does not change because it identifies flows based on the source and destination addresses, and hence does not rely on any edge-router mechanism. Hence, as long as all the core-routers support FQ, edge routers do not need to provide any support toward achieving fairness in the network. On the other hand, CSFQ and iQ rely on edge routers to convey information about flows. Thus, their performance is not optimal in the absence of fair-edge-routers. However, it can be seen that iQ performs better than CSFQ in most cases.

The improvement in fairness in iQ stems from its fairness model wherein fair-flows at fair-routers are allocated bandwidth fairly. For example, say the incoming rates at a a core-router for 3 fair-flows $f1$, $f2$, and $f3$, and one legacy-flow $f4$ were 20, 40, 50, and 70 respectively, and the corresponding "legacy" allocations were 10, 20, 30, and 40 respectively. In CSFQ, the legacy flow will not be punished because CSFQ cannot recognize packets belonging to that flow[11]. Hence, the rate allocations will potentially converge to 10, 10, 10, and 70 respectively. However, in iQ, for the same scenario, the rate allocations will converge to 20, 20, 20, and 40 respectively. Note that the above hand-worked example is not a comprehensive one, and is meant to merely shed some light on the differences between the fairness models of the two compared schemes, and provide insight into the better fairness exhibited by iQ.

C. Summary of Results

In this section, we have presented results and compared the incremental deployability of iQ, CSFQ, and FQ. We show that iQ has more incremental deployability that CSFQ. FQ exhibits good incremental deployability, but maintains per-flow state and performs per-flow processing. iQ on the other hand does not maintain any per-flow state, and does not perform per-flow processing.

V. Summary

In this paper, we have approached the problem of providing fair rate service, from the perspective of incremental deployability. In the process, we have proposed a new approach for providing fair rate allocations called *iQ*. iQ enables the incremental upgrade of an autonomous domain, and allows for the graceful co-existence of fair and legacy routers, and fair and legacy flows within an autonomous domain. While the performance of iQ is the same as that of existing fair rate allocation approaches in a fully deployed network, we show through simulations on real network topologies that, iQ performs significantly better in partially deployed scenarios.

References

[1] H. Zhang I. Stoica and S. Shenker, "Core-stateless Fair Queuing: Achieving Approximately Fair Bandwidth Allocations in High Speed Networks," in *Proceedings of ACM SIGCOMM*.
[2] N. Venkitaraman R. Sivakumar, T.-E.Kim and V. Bharghavan, "Achieving Per-flow Weighted Rate Fairness in a Core-stateless Network," in *Proceedings of ICDCS*.
[3] Lucent Technologies, "Network Quality of Service: Survey Report," .
[4] Vern Paxson, "End-to-End Routing Behavior In-ternet," in *Proceedings of the ACM SIGCOMM*, 1996.
[5] Yannick Blanpain and Raghupathy Sivakumar, "A Scalable and Deployable Architecture for Fair Rate Allocation," in *GNAN Technical Report*, 2001.
[6] http://www.cybergeography.com/atlas, "," .
[7] Gojko Babic Raj Jain, Arjan Durresi, "Throughput Fairness Index: An Explanation," in *ATM Forum Document Number: ATMForum/99-0045*.

[11] The alternative is for CSFQ to drop all un-recognizable packets, which is clearly more undesirable.