Comparing Flow-Aware and Flow-Oblivious Adaptive Routing

S. Oueslati and J. Roberts
France Telecom R&D
{sara.oueslati;james.roberts}@francetelecom.com

***Abstract*— This paper compares two approaches to realizing adaptive routing in the Internet: flow-oblivious multi-path routing using optimal end-to-end congestion control and flow-aware routing using fair queueing to share bandwidth on one or several paths. Multi-path congestion control provides ideal performance, assuming protocols can be designed to approximate the optimal fluid model. Flow-aware networking is more robust and less dependent on user cooperation and provides satisfactory performance provided use of long paths is limited in heavy traffic by applying "trunk reservation".**

I. Introduction

Internet routing protocols currently provide a single path over which the packets of any given flow must be forwarded. There is no way for the flow to avoid a congested link and, in the event of failures, routing convergence to an alternative path can be long. In this paper we consider the performance of a network where users can forward their packets over multiple paths, adapting their choice depending on current traffic levels.

The objectives are to improve performance in normal load and to enhance resilience to overloads and failures. Performance must be satisfactory for both streaming and elastic flows. We seek to realize these objectives without resorting to the complexity and doubtful efficacy of class of service differentiation or QoS routing. The two approaches we envisage differ according to whether or not the network is aware of the individual flows to which packets belong.

The present best effort network is oblivious to flows. It is theoretically possible to retain this simplicity and still meet performance requirements of streaming and elastic flows if users apply improved end-to-end congestion control protocols and the network implements particular congestion notification mechanisms, as discussed in [1]. Additionally, if users can simultaneously use several paths and congestion control is applied in a coordinated manner over all of them, the benefits of adaptive routing are realized without any further changes to the network [2], [3], [4]. It is necessary, of course, to assume users all implement the ideal congestion control protocol.

The alternative is for the network to recognize flows and enforce bandwidth sharing to realize their performance requirements. We assume users identify individual flows by appropriately labelling their packets. This can be done using the IPv6 flow label field in association with source and destination IP addresses. The envisaged lightweight implementation, based on per-flow fair queueing and implicit admission control, retains the simplicity of the best effort user-network interface [5]. If the network makes several paths available for each flow, the same flow-aware mechanisms can be used to implement adaptive and multi-path routing.

We suppose routing protocols are enhanced to enable the network to propose several possible routes for any flow. In a flow-aware network, the route followed by a flow or subflow might be chosen by a router and explicitly pinned using flow tables to store the outgoing interface for each flow identifier. If the network is flow-oblivious the user must discover these routes in some way. In this paper we suggest the use of a randomized scheme that generalizes the way traffic is split over equal cost routes in the present Internet. The same scheme can also be used in a flow-aware network and might simplify implementation of this alternative.

In this preliminary study we compare the performance of flow-aware and flow-oblivious adaptive routing on a simple triangle network assuming all flows are elastic. We first recall known characteristics of statistical bandwidth sharing. We then present simulation results for flow-aware adaptive single path routing, illustrating the use of "trunk reservation" to limit the use of long paths in heavy traffic. Finally, we compare flow-oblivious and flow-aware approaches when elastic flows can simultaneously use all available paths.

II. Bandwidth sharing performance

We first recall known results for statistical bandwidth sharing and the motivation for introducing flow-aware mechanisms in the network.

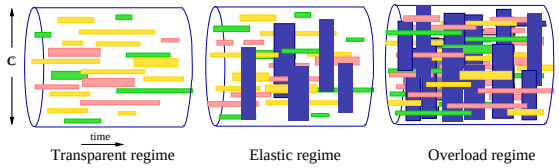

Fig. 1. Three link operating regimes

A. Bandwidth sharing regimes

For the sake of simplicity, we assume streaming flows have constant rate and elastic flows a constant exogenous rate limit due to constraints external to the considered link. Flows can then be depicted in Figure 1 as boxes whose height represents rate and whose area represents flow size. The boxes represent the flows as they would appear if the considered link were of unlimited capacity. The figure depicts three distinct regimes.

1) Transparent regime: In the left hand sketch flows share a link in a so-called transparent regime. All flow rates are relatively small and offered load ρ (flow arrival rate $\times$ average flow size / link rate) is such that the combined rate is almost always below link capacity. A small buffer is then sufficient to avoid packet loss and delays are very small so that flow impairment is negligible. Most IP backbones currently operate in the transparent regime.

2) Elastic regime: Momentary congestion occurs when some high rate flows can saturate the residual bandwidth, as depicted in the middle sketch. We assume such flows are elastic. Processor sharing models provide useful insights into the way bandwidth is shared in this case [6], [7]. The expected throughput of a bottlenecked flows is roughly $C(1-\rho)$. Note that this is typically very large so that only users with exceptionally high exogenous rate limits are concerned. Independently of the population size of such users, the processor sharing models predict that the random number of bottlenecked flows in progress is relatively very small (at most some 10s of flows) compared to the number of non-bottlenecked flows (100s of thousands of flows) [8]. Unfair sharing between concurrent flows is not usually a concern since expected throughput is largely independent of sharing weights when ρ is not too close to 1.

3) Overload regime: Independently of flow rates, severe congestion will set in if we have $\rho > 1$ for any significant length of time as illustrated in the right hand sketch. Streaming flows suffer unacceptable packet loss and elastic flow rates tend to zero. To mitigate the impact of overload, it is necessary to apply admission control to block new flows in order to protect the quality of flows in progress. Note that flow blocking on a given link may simply be the necessary signal that an alternative network path should be tried, as discussed later.

B. End-to-end congestion control

Bandwidth is currently shared under the end-to-end control of TCP. That this is generally satisfactory may be mainly due to the fact that almost all links currently operate in the transparent regime and, for most users, TCP is in fact only effective in sharing bandwidth at the network access. For flows with a sufficiently high exogenous rate to lead to the elastic regime, current versions of TCP are known to be inadequate and need to be replaced by enhanced versions.

Design of such protocols has been informed by the optimization and control theoretic framework introduced by Kelly and advanced by many researchers over the last few years (see [1] for a survey). In the fluid approximation, assuming continuous rate changes and perfect congestion notification, these protocols realize bandwidth sharing according to a desired maximum utility criterion without requiring the network to be aware of individual flows. By using active queue management and explicit congestion notification, buffer occupation can be made small enough to be compatible with the latency requirements of streaming applications.

Despite its obvious appeal, there remain significant challenges in implementing this approach:

- designing packet-based rate adjustment and congestion notification algorithms that approximate the ideal fluid behaviour;
- introducing new protocols while preserving adequate performance for users of legacy versions;
- ensuring network resilience to user misbehaviour;
- limiting the impact of flow level overload.

There is still considerable discrepancy between theoretical expectations and the realized performance of proposed protocols [9], [10]. Overload control at best relies on users testing traffic conditions at the start of a flow and backing off if the quality they require cannot be assured. This does nothing to protect the quality of possibly more exigent flows already in progress (e.g., adding more and more audio flows will eventually bring the fair rate below that required for a video flow).

C. Flow-aware networking

To enforce bandwidth sharing in the elastic regime and to avoid undue performance degradation in overload, without relying on an unrealistic degree of user cooperation, we have argued that the network should be flow-aware [5]. The minimalist approach we recommend is to identify flows “on the fly” without signalling and to implement two complementary router mechanisms: per-flow fair queueing and flow-by-flow admission control.

Fair queueing imposes max-min fairness as long as users sustain the fair share throughput of their bottleneck link. Sharing is thus independent of the particular transport protocol used allowing considerable flexibility in testing and introducing new versions. The latency of non-bottlenecked streaming flows is protected by the scheduler and is very small at normal loads. Fair queueing is scalable and feasible since only the relatively small number of bottlenecked flows actually need scheduling [8].

Admission control is necessary to avoid flow level congestion in the overload regime. By maintaining a list of flows in progress, new flows can be rejected as necessary simply by proactively discarding their packets. User applications are expected to recognize such discards as an implicit blocking signal, or as an invitation to test an alternative path, as discussed later. According to our previous work, new flows should be rejected when the throughput of bottlenecked flows goes below 1% of link capacity [11]. This represents a compromise between the risk of needlessly blocking flows when load is within capacity and the need to preserve useful throughput for admitted flows in overload. Obviously, the threshold should also be high enough to ensure the flows of a chosen range of streaming applications are not bottlenecked.

Fair queueing and admission control are mutually beneficial: the scheduling algorithm readily provides the congestion measurements necessary for admission control; fair queueing is scalable because admission control prevents the number of bottlenecked flows exploding in overload.

D. Robustness

Flow-aware networking is intended to remove some of the vulnerability due to reliance on user cooperation. It remains possible for users to cheat, however, notably by establishing multiple flows in parallel in order to gain bandwidth greater than the fair share. This seems to be an unavoidable risk. It has to be alleviated by removing the incentive to cheat. One way of doing this is to improve quality of service by allowing users to have access to more than the bandwidth of just one path.

III. Flow-aware adaptive routing

We assume each flow may be routed over just one of a set of paths designated in some way by the network. Typically, eligible paths would be of one or two hops only, each hop being a link of fixed capacity. These links might be created as paths in an underlying network using MPLS, for example.

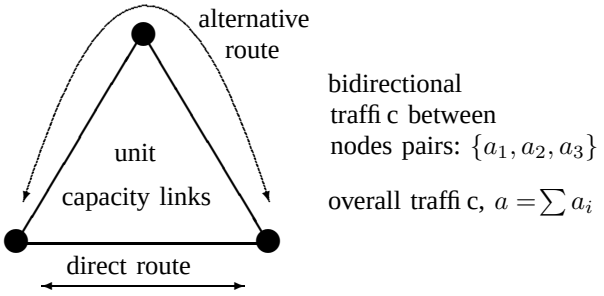

Fig. 2. Triangle network used for simulations

A. Triangle test network

To illustrate performance trade-offs in this and the next section we perform simulations for the simple triangle network shown in Figure 2. Flows are all elastic with unlimited exogenous rate. They arrive as a Poisson process with size drawn from an exponential distribution. Links are of unit capacity and traffic offered between node pairs is designated a_1, a_2 and a_3.

B. Objectives

From previous work on adaptive routing (e.g., [12], [13]), it is well-known that a flow-based adaptive routing scheme should fulfil the following objectives:

- users should be offered a wide choice of possible routes in light load in order to maximize throughput;
- in heavy traffic, on the other hand, flows should be restricted to the shortest paths to minimize blocking.

To meet these objectives we propose to apply the technique of "trunk reservation" borrowed from the telephone network: flows will not be routed over a long (2-hop) path when the bandwidth they would acquire is less than a given fraction of link capacity, θ_r.

As an overload control, flows are rejected if their throughput would be less than θ_d ($\theta_d \leq \theta_r$). In the simulations we suppose $\theta_d = 0.01$ (cf. Sec II-C).

C. Overflow routing

We first assume the network pins a flow to the shortest available path when it arrives. Specifically, we suppose the flow is routed over the direct path if its bandwidth would not be less than θ_d; if this path is congested, the flow chooses a long path with available bandwidth greater than θ_r. Results for the triangle show that performance under symmetric loading ($a_1 = a_2 = a_3$) is largely insensitive to the choice of θ_r and there is no gain over shortest path routing. Figure 3 plots blocking and throughput for node pair 1 against overall traffic $a = \sum a_i$ under asymmetric load with $a_1 = 2a_2 = 2a_3$. Trunk reservation has no significant impact

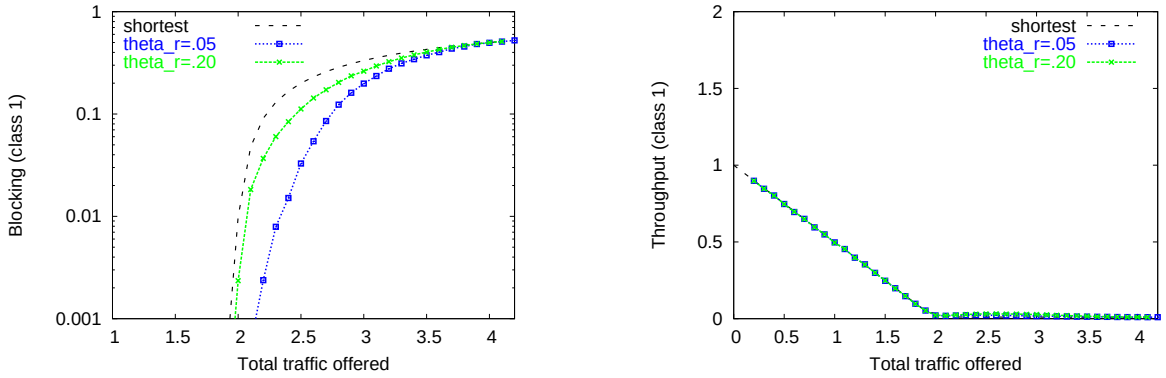

Fig. 3. Blocking and throughput performance for shortest path and overflow routing with trunk reservation parameters $\theta_r = .05$ and $\theta_r = .20$ under asymmetric traffic $a_1 = 2a_2 = 2a_3$.

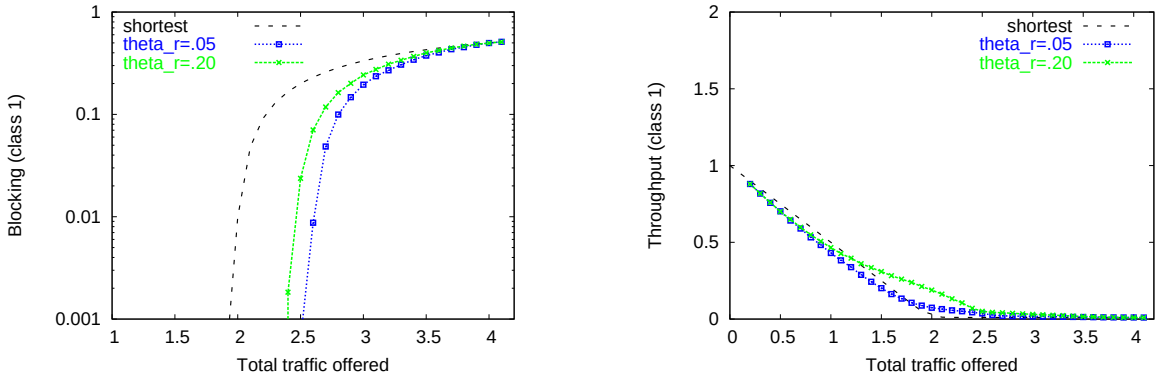

Fig. 4. Blocking and throughput performance for shortest path and randomized single path routing with trunk reservation parameters $\theta_r = .05$ and $\theta_r = .20$ under asymmetric traffic $a_1 = 2a_2 = 2a_3$.

on throughput but improves blocking performance. In this case a small value of θ_r (0.05) brings a significant improvement. This conforms to known results for trunk reservation in the telephone network where a value of θ_r between 5 and 10% is recommended [12, p.320].

D. Randomized routing

To perform load balancing over equal cost paths, routers currently choose the path by evaluating a hash function of a concatenation of source and destination IP addresses. The value of the hash function designates the path that will be followed by all the packets of the same flow. We propose here to additionally include the flow label in the hash function argument. By this device users can test several paths simply by changing the value of the flow label in the first packets of a flow. In particular, if the first chosen path is congested, the user can successively initiate new attempts with new flow labels and eventually set up the flow over an available alternative path.

In Figure 4 we show results for an assumed implementation where the two paths in the triangle are chosen with equal probability and the user settles for the first accepted attempt. Traffic is asymmetric, as in Fig. 3. In this case, a larger trunk reservation parameter is preferable ($\theta_r = 0.2$) and leads to improved throughput.

IV. Multi-path routing

Performance can be enhanced if users are able to use several paths for a single flow. In this section we seek to compare the optimal multi-path routing realized by congestion control, as per the original proposal of Kelly et al. [2] (see also [3], [4]), with an alternative realization based on flow-aware networking. We apply admission control as above to maintain the throughput of accepted flows above a threshold $\theta_d = 0.01$.

A. Congestion controlled multi-path

To discover the paths available for a given flow, we could apply the randomized choice described above: users initiate a number of subflows, all with different flow labels, thus testing a random set of paths.

The ideal fluid flow multi-path congestion control with continuous rate change, as defined by the differential equations in [2], [3], [4], is such that flows share bandwidth fairly (with respect to a chosen utility function) under the constraints imposed by the various

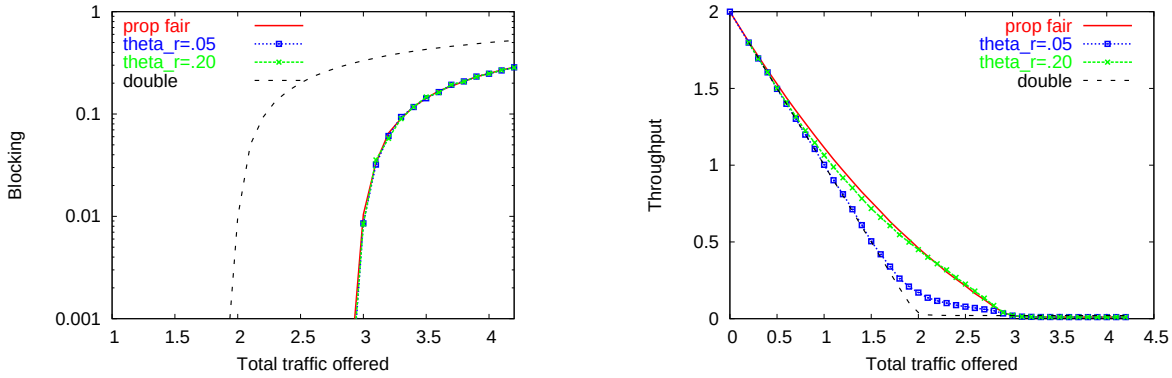

Fig. 5. Blocking and throughput performance for multi-path routing comparing proportional fairness and flow-aware routing with trunk reservation parameters $\theta_r = .05$, $\theta_r = .20$ and $\theta_r = 0.01$ (i.e., systematic double paths) under symmetric traffic $a_1 = a_2 = a_3$.

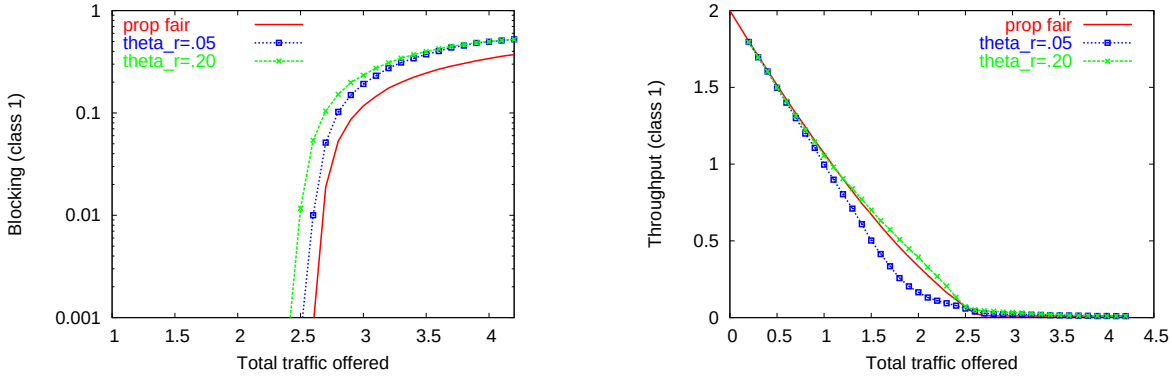

Fig. 6. Blocking and throughput performance for multi-path routing comparing proportional fairness and flow-aware routing with trunk reservation parameters $\theta_r = .05$ and $\theta_r = .20$ under asymmetric traffic $a_1 = 2a_2 = 2a_3$.

link capacities. In particular, the collective throughput of multiple subflows sharing a given link is independent of their number so that there is no requirement to ensure paths are disjoint.

On the other hand, this flow-oblivious approach has no obvious means of implementing admission control. Use of multiple paths augments the stability region and therefore makes the overload regime less likely. However, if this regime does occur, the consequence is more serious since the traffic of a greater number of node pairs is affected. In our simulations we apply admission control with the threshold θ_d without resolving how this could be performed in practice.

In the simulations we assume flows use both available routes and that the theoretical bandwidth sharing is realized perfectly and instantaneously. Let the overall rates attributed to node pair i be ϕ_i and denote by x_i the respective number of flows in progress. Simulation of the triangle network is facilitated since the ϕ_i can be determined explicitly for given (x_1, x_2, x_3) for either max-min or proportional fair sharing (see appendix). Results confirm that both types of fairness realize practically the same performance in dynamic traffic and we therefore only show results for proportional fairness. We defer comments on Figures 5 and 6 to the next section. Note that an analytical performance evaluation is possible assuming the allocation ϕ is balanced fair [14].

B. *Flow-aware multi-path*

Relying on end-to-end multi-path congestion control clearly suffers from the same implementation and robustness problems evoked in Sec. II-B. We are therefore motivated to examine the comparative performance of an alternative flow-aware networking approach.

Randomization can again be used to discover different paths. However, in the absence of coordinated congestion control, the combined rate of subflows routed on the same bottleneck link would be proportional to their number. A possible fix is to segment the flow label field: the first L_1 bits ($L_1 = 12$, say) of the label are the same for all subflows, the remaining L_2 bits ($L_2 = 8$, say) can be varied to explore the available paths. Max-min fair sharing is then realized on the path if the fair queueing algorithm uses only the L_1 field to identify its flows.

The imposition of max-min fairness by fair queueing means that it is not possible to coordinate the rate

of subflows over different network paths. If there is no restriction on path set up, network stability can be compromised due to inordinate use of long paths [15]. We therefore propose to use trunk reservation to block subflows on the long route at an intermediate level of congestion. In our simulations each flow uses the direct path if the subflow throughput is greater than θ_d and additionally uses the alternative path if it provides a subflow throughput greater than θ_r.

Figure 5 shows results for symmetric traffic. The figure compares the performance of proportional fairness, trunk reservation with $\theta_r = 0.05$ and $\theta_r = 0.2$ and systematic double routing ($\theta_r = \theta_d$). Results for the latter case illustrate the inefficiency of using two paths as traffic approaches the lower stability limit predicted in [15]. Parallel routing with a trunk reservation parameter of $\theta = 0.2$, on the other hand, provides performance equivalent to that of proportional fairness.

Figure 6 shows that, in asymmetric traffic ($a_1 = 2a_2 = 2a_3$), the parameter choice $\theta_r = 0.2$ preserves throughput but leads to somewhat greater blocking than proportional fairness. Performance is not highly sensitive to the choice of θ_r.

V. Conclusion

Adaptive routing clearly improves network performance especially in overload and failure conditions. End-to-end multi-path congestion control according to the theoretical optimal fluid model, would lead to ideal performance. However, implementation is not straightforward and performance remains vulnerable to misbehaving users or wrongly configured terminals. Flow-aware networking based on fair queueing and admission control mechanisms appears as an more pragmatic alternative. It brings comparable performance benefits provided trunk reservation is applied to limit use of long paths in heavy traffic. Our results suggest the reservation parameter should be around 20%, considerably higher than the preferred value in telephone networks.

Though these results are encouraging, it is clear that the present evaluation is very limited in scope and further investigation is necessary. We intend in future work to consider more complex networks offered a realistic traffic mix with a range of flow rates.

Appendix: $\phi(x)$ for proportional fairness and max-min fairness for the triangle

Capacity constraints are $\phi_i + \phi_j \leq 2$ for all index pairs (i, j). Note that these are the same capacity constraints as for a triangle with link capacity 2 where flows are routed over the long path only. The allocations are as follows:

Proportional fair

if $(x_1 > x_2 + x_3)$
$\phi_1 = 2x_1/(x_1 + x_2 + x_3)$
$\phi_2 = \phi_3 = 2 - \phi_1$
and symmetrically for other index permutations;
if no x_i is greater than the sum of the others,
$\phi_1 = \phi_2 = \phi_3 = 1$.

Max-min fair

if $((x_1 + x_2) \geq (x_1 + x_3)$ and $(x_1 + x_2) \geq (x_2 + x_3))$
$\phi_1 = 2x_1/(x_1 + x_2)$
$\phi_2 = 2 - \phi_1$
$\phi_3 = \min(\phi_1, \phi_2)$
and symmetrically for other index permutations.

References

[1] S. Low and R. Srikant, "A mathematical framework for designing a low-loss, low-delay internet," *Network and Spatial Economics, vol. 4, no. 1, March 2004, pp. 75-102*, 2004.

[2] F. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society, vol 49, pp237-252*, 1998.

[3] T. Voice, "Delay stability results for congestion control algorithms with multi-path routing," *Stats Lab report 2004-11, University of Cambridge*, 2004.

[4] H. Han, S. Shakkottai, C. Hollot, R. Srikant, and D. Towsley, "Multi-path tcp: A joint congestion control and routing scheme to exploit path diversity on the internet," *IEEE/ACM Transactions on Networking, to appear*, 2005.

[5] S. Oueslati and J. Roberts, "A new direction for quality of service: Flow aware networking," *NGI 2005, Rome*, April 18-20, 2005.

[6] T. Bonald and L. Massoulié, "Impact of fairness on internet performance," *Proceedings of ACM Sigmetrics*, 2001.

[7] T. Bonald, L. Massoulié, A. Proutière, and J. Virtamo, "A queueing analysis of max-min fairness, proportional fairness and balanced fairness," *Queueing Systems, to appear*, 2006.

[8] A. Kortebi, L. Muscariello, S. Oueslati, and J. Roberts, "Evaluating the number of active flows in a scheduler realizing fair statistical bandwidth sharing," *Sigmetrics'05, Banff, Canada*, June 2005.

[9] Y-T. Li, D. Leith, and R. Shorten, "Experimental evaluation of tcp protocols for high-speed networks," *Technical report, Hamilton Institute, NUI Maynooth, Ireland*, 2005.

[10] L. Xu, K. Harfoush, and I. Rhee, "Binary increase congestion control (bic) for fast long-distance networks," *Proceedings of IEEE Infocom 2004*, 2004.

[11] N. Benameur, B. Ben-Fredj, S. Oueslati-Boulahia, and J. Roberts, "Quality of service and flow level admission control in the internet," *Computer Networks 40 (2002) 57-71*, 2002.

[12] G. Ash, *Dynamic routing in telecommunications networks*, McGraw-Hill, 1998.

[13] S. Oueslati-Boulahia and J. Roberts, "Impact of "trunk reservation" on elastic flow routing," *Proceedings of Networking 2000, Springer LNCS vol 1815, pp 823-834*, 2000.

[14] Juha Leino and Jorma Virtamo, "Insensitive load balancing in data networks," *Computer Networks*, 2005, to appear.

[15] P. Key and L. Massoulié, "Fluid models of integrated traffic and multipath routing," *Queueing Systems, to appear*, 2006.