Combined Multipath Routing and Congestion Control: a Robust Internet Architecture

Peter Key, Laurent Massoulié, Don Towsley

***Abstract*— Network management is complicated by uncertain traffic patterns and workloads. Flexible routing schemes mitigate some of the problems by making the exact location of capacity less important: if there is available capacity the routing scheme will find it. In this paper we propose a combined multipath routing and congestion control architecture that gives performance improvements to the end user and simplifies network dimensioning. We advocate multihoming and stepping stone routers to provide path diversity, and a congestion controller and path selection algorithm that automatically balances traffic across the lowest cost paths. A notable feature of a multipath congestion controller is that it cannot be tuned to a single RTT, hence it differs from standard TCP with respect to RTT bias. Scalability of the architecture results from implementing the algorithms at end-systems. We illustrate on network topologies of interest the performance impact of our architecture: active use of two paths can (i) halve response times and (ii) double the load that a network can carry.**

I. INTRODUCTION

Network design and management are challenging problems and raise questions regarding how to choose topology, link bandwidths, and routes as well as how to quickly identify and react to failures. Answers to these questions are further made difficult due to uncertainties in traffic workloads. These questions are receiving increasing attention. For example, [15], [16], [12] propose the use of randomized route selection for the purpose of making traffic workloads within a single network more predictable and to provide balanced loads within the network. These proposals achieve robustness against traffic workload changes at the cost of potentially doubling the necessary bandwidth. Similar ideas have been proposed to speed up recovery from link/path failure, [5]. None of the above proposals are able to make failures entirely transparent to the application.

In this paper we propose a new Internet architecture within which nodes are multi-homed and each session initiates a data flow that is spread over multiple paths by sending distinct packets over potentially distinct paths, and controlled by a single entity. In our proposal, the set of paths is continuously improved by some random selection mechanism, similar to that in [5], where one path is the direct path between source and destination. This path can ensure efficient resource usage during normal operation whereas the randomized paths provide network-level load balancing during abnormal times. Because several paths are maintained throughout the duration of the session, a failure affecting one path is never visible to the user.

Our architecture couples recent advances in the design of multipath congestion controllers, [14], [6], [8] with the potential of multi-homing and randomized route selection to reduce the complexity of network management while at the same time improving performance and robustness to the user. We show through simple models that the performance (in terms of response of a session) within this network can be at least twice as good as in the current Internet, and the network may carry twice that which could be achieved within the network proposed in [15], [16], [12]. We find that most of this benefit can be achieved using no more than two paths.

Besides the need for a coordinated congestion controller, a mechanism is required for setting up random routes. We present a simple source routing algorithm based on the use of *stepping stone (SS)* routers scattered throughout the Internet. This is similar to the scheme presented in [12] except that it is Internet-wide rather than limited to a single backbone.

An application level proposal related to ours [3] proposes the use of Web proxies in an overlay to improve the performance for Web clients. In contrast, we propose a solution that is application independent, and which gives the end-node (the source) greater responsibility in choosing the paths that it should use. One implication of the end-system approach is that it potentially changes some of the economic incentives for both end-users and ISPs.

The remainder of this paper is organized as follows. In Section II we present the architecture in greater detail, focussing on the multipath congestion controller and the path selection algorithm. Section III compares the proposed architecture to the existing Internet and to the architecture proposed in [15], [16], [12] both from the perspective of performance and reliability. It also suggests that two paths are generally sufficient. Section IV focuses on interim measures. These include the use of multiple TCP connections, one on each path instead of a single multi-path controller, Section V concludes by discussing some implementation issues.

II. ARCHITECTURE

The key ingredients of our architectural proposal are firstly, diversity, which is achieved through a combination of multi-homing and random path sampling, and secondly route selection and multipath streaming using a congestion controller that actively streams along the best routes from a working set[1]. We now expand on these concepts.

A. Multihoming

Many authors have commented that multihoming is a way of providing both resilience and performance improvement, but studies have been limited by the implementation and addressing issues associated with IPv4. In terms of availability, although home-users are currently often limited in their choice of ISP, in contrast campus or corporate nodes may have diverse connections, via different ISPs. Moreover the growth of wireless hotspots, wireless mesh and broadband wireless in certain parts of the globe means that even home users may become multi-homed in the future. Recent figures [2] suggest that 60% of stub-ASes (those which do not transit traffic) are multihomed, and [13] claims that with IPv6 type multihoming there are at least two disjoint paths between such stub-ASes. The study in [1] discusses performance improvements for multihoming in the current (IPv4) Internet. Multihoming requires several addresses per end-system, which is made possible by IPv6. Note that the addressing issues of multihoming are being discussed in the IETF, in both the IPv6 (in the multi6 working group) and mobile IP contexts.

B. Stepping-stone routers

Multihoming goes some way towards addressing the critical issues of having diverse paths: essentially for both performance and reliability reasons we would like at least 2 disjoint paths between source and destination. See also [5] for empirical evidence that four paths are typically enough for failure recovery. In addition, for efficiency reasons we would like to be able to spread load across a number of different paths, possibly even within a single AS. Hence we also need stepping-stone routers acting as intermediary nodes, through which we can route. A number of authors eg [5] have considered one-hop source routing, which routes to some intermediate node (router) which then forwards the packets to the destination. Our proposal is in this spirit. The control can be end-system based, where the source only sends one of the destination addresses to the stepping stone router, thereby choosing the ingress link, so that the stepping-stone router then just acts as a forwarding engine.

The SS-routers themselves could be advertised via a new DNS-like service, where a stepping-stone router is returned along with the IP address, based on the source address and destination address or addresses sent by the source. We could envisage a set of stepping-stone routers being returned. Such stepping stone-routers could also be implemented using multiple home agents in the context of mobile IP. Note that within an AS, the use of such routers means that the end-system (node) is potentially performing the load-balancing, rather than the AS itself. However, assuming that the AS itself is in charge of advertising the SS-routers returned by the query, it couldselect the set of stepping-stone routers based on its own load measurements, and vary the choice over time.

C. Route selection and coordinated congestion control

We perform load balancing at two levels. At the lower level, the Coordinated Multipath Congestion controller actively balances the load across a given set of paths, thereby splitting the session load across the lowest cost paths. It is thus different from the load-balancing algorithms of both [15] and [5] which split load to paths according to fixed splitting ratios, irrespective of changes in traffic conditions. It operates on a time scale dictated by the RTT's of the current paths. At the higher level, we periodically resample for new paths, which is done via random selection of stepping stone routers. This is done at a time scale of the order of seconds or minutes.

A more specific description of the congestion controller is as follows. We associate a utility function with a congestion controller [7]. The controller shifts its load to the paths with the lowest loss rates (or more generally with lowest ECN mark rate or largest delays) and equates the marginal utility of its aggregate data rate to the loss rate on these "best" paths. For example, TCP can be thought of as implicitly using a utility function of the form $U(x) = -w/x$, where w is some weight and x is the rate of the connection; in the case that the weight w is given by $w = 1/(RTT)^2$, equating the derivative of the utility to the path loss rate p produces the familiar relation $x = \frac{1}{RTT\sqrt{p}}$.

Note that there is a fundamental issue for TCP-friendliness here, caused by the current round-trip time bias in TCP: for our coordinated controller we need a single utility function, hence a single weight, which implies a common value of the RTT. This could be an average value, or maximum RTT for example. More radically one could consider removing RTT bias altogether.

Recent research [14], [6], [8] has shown it is possible to design efficient multipath controllers that rely only on local path information. Such controllers are TCP-like: for each path there is a steady increase of the rate and

---

[1] Throughout the paper, by path we mean a concatenation of links.

a decrease which is related both to the feedback signals from the path (eg loss events) and the rate of aggregate acknowledgements from *all* the available paths.

Route selection is used to continuously search for low cost paths. We suggest the following implementation. The congestion controller aims to use a fixed number of paths (eg two) per nominal "route", i.e. per distinct source-destination address pair. For instance, a dual-homed source routing to a single-homed destination would aim to use 4 paths (in this example, such paths would not be disjoint as they share the last hop to the destination). The congestion controller periodically chooses a new stepping-stone router at random per nominal "route", and adds the corresponding path to the set of paths currently used. After a probing phase, which can be done in band using actual data, the controller suppresses the path that received the poorest performance (reflected by the loss rate, for example) from the set of active paths, thus returning to the desired number of active paths per nominal "route". The fact that this is end-system driven avoids the scalability problems of other proposals, e.g. [3].

To illustrate multipath sharing, consider the simple network of Figure 1. There are two resources with respective capacities C_1, C_2, n data transfers use multipath congestion controllers and use the two resources at rates x and y respectively, while m transfers use only resource two, at some rate z. The allocations x, y, z are characterised as solutions to the optimisation problem:

$$\begin{aligned}&\max\{nU(x+y)+mU(z)\}\\&\text{subject to } nx \le C_1, \quad ny+mz \le C_2,\\&\text{over } x,y,z \ge 0,\end{aligned}$$

for some utility function U, assumed common to the two types of transfers for simplicity. Routine calculations show that the resulting shares are given by $x = C_1/n$, $y = 0$, $z = C_2/m$ provided $C_1/n \ge C_2/m$, and by $x = C_1/n$, $x+y = z = (C_1+C_2)/(n+m)$ if $C_1/n < C_2/m$. One special case of interest is when there are no transfers of the second type ($m = 0$): then the n transfers all get a share $(C_1 + C_2)/n$ of the aggregate capacity. This is to be contrasted with a single path scenario where type 1-transfers access resource 1 only, resulting in shares $x = C_1/n$ and $z = C_2/m$ respectively.

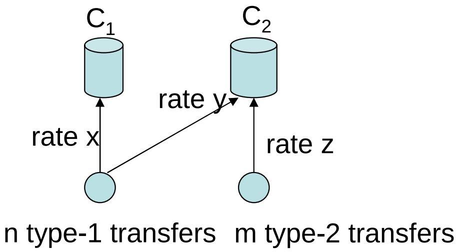

Fig. 1. Example of a network with multipath connections

III. ARCHITECTURE EVALUATION

A. methodology and general properties

We evaluate our architecture by identifying the traffic demand that can be supported with our schemes in comparison to alternative schemes. We illustrate our methodology using the example of Figure 1, for which we had two types of data transfers.

We associate a demand, or load, with each type, by specifying a corresponding arrival rate of transfer requests, ν, and the average amount of data requested per transfer, S. These two parameters define the demand rate, or load $\rho = \nu S$ for the corresponding class. A more detailed demand model is obtained by assuming transfer requests arrive at the instants of a Poisson process with rate ν, and the requested amounts of data are i.i.d., exponentially distributed with mean S - in effect we are using a session model for which Poisson arrivals have both a theoretical and empirical justification. We then consider that the architecture can cope with the loads ρ_1, ρ_2 if the corresponding stochastic process reaches a stationary state. In the example of Figure 1, this happens provided $\rho_1 < C_1 + C_2$ and $\rho_2 < C_2$. A general method for proving this is to consider differential equations describing the mean evolution of the number of flows in progress, given for our example by

$$\frac{d}{dt}n = \nu_1 - \frac{1}{S_1}n(x+y), \quad \frac{d}{dt}m = \nu_2 - \frac{1}{S_2}mz,$$

and establish that their solutions converge to zero; see e.g. [4] for further details.

Our architecture makes efficient use of the network resources for general network topologies[2]. A formal statement of this property, proved in a longer version of this paper, is as follows.

Theorem 3.1: Assume that class r-transfers can use any network paths from an associated set $\mathcal{P}_r$ (via a combination of multihoming and stepping stone routers). If there exists some split of the load ρ_r of class r-transfers into path loads ρ_{rp}, $p \in \mathcal{P}_r$ such that the network resources can carry the path loads ρ_{rp}, then the architecture based on multipath congestion controllers and random path resampling from the available set $\mathcal{P}_r$ will effectively find such a feasible split and hence carry the loads ρ_r.

Moreover in a model incorporating link cost functions, at equilibrium the load is split in such a way as to minimize the aggregate cost *irrespective* of which utility functions congestion controllers implement [9].

---

[2] The general model of a topology we consider is a convex, non-increasing set of vectors of path loads. For detailed descriptions and further examples see [9].

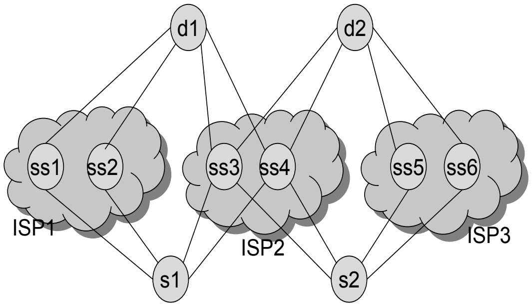

Fig. 2. Multihoming with stepping stone routers. Example with two sources $s1$, $s2$ two destinations $d1$, $d2$, all dual-homed, and two stepping-stone routers per AS.

B. Dual-homing

We first illustrate how our architecture performs on a simple example with dual-homed users, and a simple model of an AS. Specifically, consider a source destination pair that connect through two different AS's, 1 and 2. For the sake of this example, assume that the bandwidths over the two paths are C_1 and C_2. We are interested in the traffic load that different transport schemes can support.

Traditionally, the path used is chosen due to some *policy*, e.g., the Abilene network in the USA only allows access to traffic destined to academic institutions, or one path is used as a backup in case the other fails. Let us assume that the policy in place permits fractions f_1 and f_2 over each path, $f_1 + f_2 = 1$ under normal conditions (normally either $f_1 = 1$ or$f_2 = 1$). For a total load ρ, the corresponding path loads are $\rho_1 = f_1\rho$, $\rho_2 = f_2\rho$. Then the current Internet architecture supports loads satisfying $\rho_1 < C_1$, $\rho_2 < C_2$ or equivalently

$$\rho < \min\{C_1/f_1, C_2/f_2\}.$$

On the other hand, if each session uses the multipath congestion controller described in the previous section, then the multipath architecture can support loads satisfying

$$\rho < C_1 + C_2.$$

Note that the current architecture can achieve this load provided that $f_1 = C_1/(C_1 + C_2)$. Although such a choice of f_1 is possible, in most cases policy dictates against it.

Consider a scenario where one of the paths fails. In the current Internet, this may cause active sessions to time out. On the other hand, the multipath architecture reacts gracefully to the failure.

Some link failures are not catastrophic in that all traffic is prevented from traversing a path. In such cases we assume that the capacity of the affected path, say path 1, is degraded from C_1 to $C'_1 < C_1$. In this case, the resulting capacity region depends on the architecture. In the current Internet the capacity reduces to

$$\rho < \min\{C'_1/f_1, C_2/f_2\},$$

whereas in the multipath case, it reduces to

$$\rho < C'_1 + C_2.$$

Consider for concreteness the case where $C_1 = C_2 = C$, and $f_1 = f_2 = 0.5$ in a static routing scenario. Then, writing $\Delta C = C - C'$, the capacity reduction after a single link failure, we find that static routing can carry $\rho = 2C - 2\Delta C$, i.e. an effective capacity loss of $2\Delta C$, while multipath can carry $\rho = 2C - \Delta C$, thereby incurring half the capacity loss of static routing.

Conversely, assume that the traffic splitting ratios f_1, f_2 are enforced on the basis of a hash of the per transfer IP source/destination addresses, and have again nominal values $f_1 = f_2 = 0.5$. A shift in demand can then be modelled as a deviation of f_1 and f_2 from these nominal values. Such shifts in demand have no impact on the capacity of multipath; however the capacity of static routing is affected by any such shift. When $C_1 = C_2 = C$, its capacity goes down from $2C$ to $C/(0.5 + \epsilon) \approx 2C - 4\epsilon C$ when f_1 goes from 0.5 to $0.5 + \epsilon$ for some small ϵ.

As we have just seen, multipath congestion control is robust to shifts in demand or in capacity, in contrast to static single path load balancing. In fact, even with fixed capacities and demand parameters, multipath is beneficial.

Indeed, consider the case where $C_1 = C_2$, and each of the two resources receives, under static single path load balancing, a load of $\rho = \nu S$, with $\rho < C$. Then, for the stochastic model of demand above described, each resource in isolation is an M/M/1 processor sharing queue. Standard results [11] show that the mean transfer time equals $S/(C - \rho)$. Now consider the case where each transfer uses simultaneously the two resources, thus proceeding at a speed of $2C/n$ when there are n active transfers. This is again an M/M/1 processor sharing queue, but with doubled capacity and doubled load. Thus, the mean response time now reads $S/(2C - 2\rho)$, half of the value achieved without multipath routing. This illustrates the point that multipath routing achieves higher levels of statistical multiplexing.

A natural extension of this simple model of dual-homed users is as follows. Transfers of a given type r can access a specific set $\mathcal{P}_r$ of resources, and each resource ℓ has an associated capacity C_ℓ. Type r transfers contribute a load ρ_r to the system. This is a generalisation of Figure 1, which covers e.g. the topology of Figure 2 if the bottlenecks are at the stepping stone routers.

In this set-up, resources are sufficient to cope with demand provided for any collection $\mathcal{S}$ of types r, the

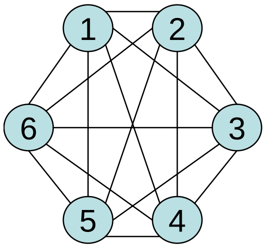

Fig. 3. A full mesh topology

corresponding aggregate demand is less than the total capacity accessible to that demand, i.e.

$$\sum_{r\in\mathcal{S}}\rho_r \le \sum_{\ell\in\cup_{r\in\mathcal{S}}\mathcal{P}_r} C_\ell. \tag{1}$$

We consider a specific instance to illustrate resilience to overload. Assume there are L resources in total, each having capacity C, and m sources, each of which generates traffic at rate ρ, where $2C > \rho > C$. We assume each source can use one dedicated resource, as well as a fixed randomly selected additional resource. We might think of this as an overload scenario with m overloaded sources.

It can then be shown that the probability that the random selection of alternative resources is unable to carry the offered load is of order m^2/L. So using just a single fixed alternative without resampling will succeed provided no more that $o(\sqrt{L})$ sources are overloaded.

If we can choose the alternate routes with care, then we will be able to satisfy all the demands provided $m < L/2$. This will in effect be the case with resampling. This will also be the case with stochastic arrivals if each new flow chooses a random alternative when it arrives, thus spreading the load across the resources.

C. Mesh topology

Consider now the full mesh topology illustrated in Figure 3. There are N nodes, and a directional link with capacity C between any ordered node pair. This topology has been considered in [15], [16] as a candidate backbone architecture. The authors in [15] consider the following mechanism –the so-called Valiant Load Balancing (VLB)– for efficiently using the available capacities in such a network. Traffic demand from node i to node j is split into $N-1$ equal shares. One share is routed via the direct $i-j$ connection, while the $N-2$ remaining shares use a two-hop path $i-k-j$, with k spanning the remaining $N-2$ nodes.

The authors in [15] show that for any traffic matrix $(\rho_{ij})_{i,j\le N}$ such that for each node i, both the outbound traffic $\rho_{i\cdot} := \sum_j \rho_{ij}$ and the inbound traffic $\rho_{\cdot i} := \sum_j \rho_{ji}$ are less than some per-node pre-specified capacity r, then a per-link capacity $C = 2r/N$ suffices to carry the traffic.

In the present scenario, according to our scheme transfers from node i to node j would proceed along two simultaneous paths taken from the set of direct or two-hop paths, with continuous re-sampling to identify efficient paths. Theorem 3.1 above implies that any traffic matrix (ρ_{ij}) such that the loads ρ_{ij} can be split into path loads, ρ^0_{ij} for the direct path, and ρ^k_{ij} for the path $i-k-j$, such that the link capacity constraints are satisfied, i.e. for all i, j,

$$\rho^0_{ij} + \sum_{k\neq i,j} \rho^j_{ik} + \rho^i_{kj} < C,$$

then our scheme is able to cope with the demand, and effectively finds such a feasible load allocation. In particular, it can cope with any demand that is handled with VLB.

It can also cope with demands that cannot be handled by VLB. Indeed, consider the case of fully symmetric demands, $\rho_{ij} \equiv \rho$ for all i, j. Then, by carrying the traffic from i to j on the direct path only, it is possible to carry loads ρ whenever $\rho < C = 2r/N$. In contrast, under VLB, loads ρ up to $r/(N-1)$ only can be handled, hence a reduction in the load that can be supported by a factor of $2(1-1/N)$.

IV. INTERIM MEASURES

We now describe possible interim measures and performance implications. The first one consists of deploying multipath congestion control, but without implementing path re-sampling. Let us illustrate the resulting performance on the mesh topology. Assume in that case that any transfer initiated from node i to node j will proceed on both the direct $i-j$ path, and on a two-hop path $i-k-j$, where k is chosen uniformly at random from the $N-2$ possible mid-nodes. That is to say, the load ρ_{ij} is split into $N-1$ equal shares, each of which is to be spread between the direct path and a two-hop path.

Then by Theorem 3.1 (or the results of [6], [9]) multipath routing can cope with the demand whenever these equal shares $\rho_{ij}/(N-1)$ can be split between their two possible paths. In particular, whenever VLB can carry the demand, then so can the above multipath routing without re-sampling. Furthermore, if traffic is symmetric, i.e. $\rho_{ij} \equiv \rho$ for all $i \neq j$, then multipath routing can exploit the direct paths, and hence cope with values of ρ as high as $C = 2r/N$, just as in the case with path resampling. Thus for symmetric loads, this again improves upon VLB by a factor of $2(1-1/N)$.

The next interim measure consists in having data transfers proceed along several paths, but now the data rate along each path is achieved by an individual congestion controller, say TCP, without coordination between such controllers. We now illustrate how this would perform on two topologies of interest. The first one is the topology with parallel resources: type r transfers use parallel, uncoordinated connections via each resource in the associated set $\mathcal{P}_r$. Then, as shown in [9], the system is able to carry offered loads ρ_r so long as Condition (1) is met with strict inequality. Thus for this topology, there is no reduction in the traffic that can be carried.

In contrast, consider now the mesh topology, with symmetric demand $\rho_{ij} \equiv \rho$. Assume as before that each data transfer from i to j proceeds in parallel along the direct $i - j$ link, and along a randomly selected two-hop path $i - k - j$. The situation is now more complex: the maximal load ρ that the network can cope with will typically depend on the congestion controller implemented along each path. For a standard TCP controller, assuming further that the loss rate along a two-hop path is just the sum is the individual link loss rates, it can be shown (see [9], proposition 3) that the system is stable provided $\rho/C < \sqrt{2} \approx 0.71$, hence a 30% efficiency loss compared to coordinated multipath.

Finally, stepping-stone routers could be implemented as a separate infrastructure, comprising an overlay, much as in [5], [3].

V. CONCLUDING REMARKS

One benefit of our approach is that by making the routing more robust, and pushing control to the edge of the network, there is less dependence on the core of the network. In other words, there is robustness against failures and performance issues within the network, since the edge is able to decide how to optimise and route around problems. With the current Internet often driven by hidden incentives (for example BGP has to make use of partial information about true costs), schemes such as ours may encourage a more overt incentive-compatible behaviour from ISPs. For example, with a coordinated congestion controller choosing routes on the basis of performance, those stepping stone-routers which offer the best performance will be chosen. If charging by (transit) ISPs is ultimately driven by carried traffic, which follows from strict per-byte charging but could be mediated by less direct means such as stratified price structures, then those ISPs which offer the best performance will receive the most traffic, coupling performance with price. The coordinated controllers may then need to be adapted to account for this, by factoring cost into the control algorithm.

The discussion so far has been based around an Internet wide solution, however the proposed solution would work equally well within a domain (where it could be seen as an alternative to VLB routing), or within an overlay, where now the stepping-stone routers would be members of the overlay (as in [5]), or within a mesh network.

There are a number of implementation choices that are yet to be made. For example, a coordinated congestion controller needs to be in some sense RTT independent, the multihoming fits better in a IPv6 scenario than v4, the granularity of the controller and the resampling needs further investigation (eg for very short flows there is no point is spreading the load) and there is question of at what layer to implement the controller. By making the solution application independent, we have to take care of issues such as packet reordering and latency for real-time. In addition, stepping stone routers need to be advertised. However none of these are show-stoppers; indeed we have described interim measures, such as using parallel TCPs rather than coordinated control, which provide some of the benefits of the approach.

REFERENCES

[1] A. Akella, B. Maggs, S. Seshan, A. Shaikh and R. Sitaraman, "A Measurement-Based Analysis of Multihoming", ACM SIGCOMM 2003, Karlsruhe, Germany.

[2] S. Agarwal, C. Chuah , R.H. Katz, "OPCA: Robust interdomain policy routing and traffic control", In IEEE Openarch (New York, NY, April 2003).",

[3] D.G. Andersen, H. Balakrishnan, F. Kaashoek and R.N. Rao, "Improving Web availiability for clients with MONET", NSDI 2005.

[4] T. Bonald and L. Massoulié, Impact of fairness on Internet performance. In Proceedings of ACM Sigmetrics 2001.

[5] K.P. Gummadi, H. Madhyastha, S.D. Gribble, H..M. Levy, D.J. Wetherall. "Improving the reliability of Internet paths with one-hop source routing", *Proc. 6th OSDI*, 2004.

[6] H. Han, S. Shakkottai, C.V. Hollot, R. Srikant, D. Towsley. "Overlay TCP for Multi-Path Routing and Congestion Control", submitted for publication.

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

[8] F. P. Kelly and T. Voice, Stability of end-to-end algorithms for joint routing and rate control, Computer Communication Review **35:2** 5–12, 2005.

[9] P. Key, L. Massoulié, "Fluid limits and diffusion approximations for integrated traffic models", Microsoft Research Technical Report TR-2005-83. Available at ftp://ftp.research.microsoft.com/pub/tr/TR-2005-83.pdf

[10] P. Key, L. Massoulié, A. Bain, F. Kelly. "A network flow model for mixtures of file transfers and streaming traffic". ITC18, Berlin, 2003.

[11] L. Kleinrock, *Queueing Systems* 1-2, Wiley, 1975/76.

[12] M. Kodialam, T.V. Lakshman, S. Sengupta. "Efficient and robust routing of highly variable traffic," HotNets 2004.

[13] C. de Lanois, B. Quoitin and O. Bonaventure, "Leveraging Internet path diversity and Network performances with Ipv6 Multihoming", Universite catholique de Louvain Tech. report RR 2004-06.

[14] W.-H. Wang, M. Palaniswami, S.H. Low. "Optimal flow control and routing in multi-path networks", *Performance Evaluation*, **52** (2003) 119-132.

[15] R. Zhang-Shen, N. McKeown. "Designing a predictable Internet backbone network", HotNets 2004.

[16] R. Zhang-Shen, N. McKeown. "Designing a predictable Internet backbone with Valiant load-balancing", IWQoS 2005.