Combining Multipath Routing and Congestion Control for Robustness

Peter Key
Microsoft Research
7 J J Thomson Aveue
Cambridge, CB3 0FB, UK

Laurent Massoulié
Microsoft Research
7 J J Thomson Aveue
Cambridge, CB3 0FB, UK

Don Towsley
Department of Computer Science
University of Massachusetts
Amherst, MA 01003-4610

***Abstract*— Flexible routing schemes mitigate some of the problems associated with uncertain traffic patterns and workloads 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 can provide performance improvements to the end user and simplifies network dimensioning for operators. We describe a flow-level model, able to handle streaming and file transfer traffic, with stochastic arrivals, and look at a fluid limit. We describe a congestion controller and path selection algorithm that automatically balances traffic across the lowest cost paths, and we suggest ways in which just two paths may be used, with a random selection policy. 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. We show that under certain conditions the allocation of flows to paths is optimal and independent of the flow control algorithm used. Scalability of the architecture results from implementing the algorithms at end-systems. We illustrate by examples how such an approach can halve response times and double the load that a network can carry.**

I. Introduction

Robustness against traffic variations or against changes in network capacity or topology is important both for network operators and end-users. A robust network allows operators to hedge against uncertainty and hence save costs, or provide performance benefits to end users.

We shall consider a particular form of robustness that is achieved by introducing diversity or multipath at the transport level, and using flow control to automatically balance across paths. The two concepts have been suggested separately, however there is added benefit in combining them, which we explore in this paper. 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].

Research associated with traditional telecommunication networks and queuing networks has looked at related questions, with a history going back 25 years, often classified as dynamic routing and resource pooling. Resource pooling means we consider sets of resources, rather than individual resources, with a concomitant gain in efficiency, and dynamic routing enables demands to access these resources. This research has received relatively little attention recently and ideas from it are only gradually reappearing in the context of modern communication networks.

The motivation for this work is a (fast packet) network such as the Internet, although the framework is more general and applies whenever there is a notion of flows through the network, which require service from a set of resources associated with some path through the network. The demand (volume) of such flows is stochastic, and the flow arrival process is also stochastic. Two examples are (i) end-user flows from a host or site, with a flow identified by some function of the typical IP 5-tuple (comprising source /destination address and port number and protocol type), (ii) aggregate intra-domain flows for an ISP, labelled by ingress and egress, or ingress and egress prefix to allow for multiple exits.

It is useful to characterise flows into two classes, *elastic* or file transfer traffic, and *streaming* traffic. Elastic traffic has a given volume to transfer, and streaming traffic has a given duration (holding time); for elastic traffic the volume may be random but is independent of the network conditions, whereas for steaming the same holds for the duration.

For a fixed number of flows, we can frame the allocation problem as an optimisation at the flow-level, creating a form of welfare maximisation by associating utility functions with individual flows, with some form of cost or penalty function associated with resources. A utility function can be thought of a capturing the essence of an underlying rate-control or flow-control scheme, such as TCP. Under certain assumptions, this is a well-posed convex optimisation problem with a unique optimum. By then considering the dynamics associated with the stochastic demand matrix, that is flows arriving as a point process, we can show that the dynamics are asymptotically stable in the sense of Lyapunov.

We require flows to be able to coordinate their flow-control across the routes they can choose from: this has some practical implications which we comment on. In particular, for the greatest benefit it is necessary to associate a single utility function with a flow, implying a single round-trip time to be used across all a flows route choices. If we do this, we can show that at one level the routing efficiency is insensitive to the utility function use. We also do not consider finer grained dynamics, relying on recent work or [1], [2] to show that

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

stability and convergence of flow-control is possible.

A further practical issue is the choice of route-sets to choose from: the optimisation assumes some performance knowledge about all the route-sets open to a particular flow, which has to be gained by some form of probing or feedback. Rather than looking across all possible route choices, we discuss ways of limiting the choice, either by periodically reselecting a small number, or by using a random or sticky-random strategy. For example, we may choose to keep one nominal route, and select one other at random form a possible set, reselecting when performance drops below some threshold. This is inspired by ideas of Dynamic Alternative Routing [3] and power of two random choices [4]. It is also attractive since the simplest and most common form of multihoming is dual homing.

We comment on these issues and questions of timescale in this paper, where the aim is to suggest first steps rather than complete solutions.

II. Resource Pooling, Diversity and Resilience

Resource pooling means treating a set of resources as one, for performance or efficiency reasons. As a simple example, motivated by dual-homing, suppose that we have two resources, each of capacity C, and each offered elastic traffic (as a concrete example, we may think of file transfers or Web traffic). Suppose further that this elastic demand arrives according to a Poisson process with rate ν, and with an associated volume distributed as a exponential random variable, with mean $F = \mu^{-1}$, producing an offered load $\rho = \nu/\mu$. The volume may be transferred as a sequence of packets, using a flow control algorithm. If the flow-control enforces perfect sharing, then each resource in isolation behaves as an $M/M/1$ processor sharing queue. Standard results [5] show that the mean transfer time equals

$$\frac{F}{(C-\rho)} \tag{1}$$

and the expected response time to transfer a demand of size f is $f/(C-\rho)$. Now consider the case where each transfer can simultaneously use the two resources, and so proceeds at a speed of $2C/n$ when there are n active transfers. This is again an M/M/1 processor sharing queue, but with twice the capacity and twice the load. The mean response time now reads $F/(2C-2\rho)$ half of the value achieved without resource pooling, and the expected response time of demand size f is also halved. This illustrates the point that multipath routing achieves higher levels of statistical multiplexing.

There is empirical evidence that Poisson arrivals is an appropriate model for session arrivals [6], and other evidence that the size of documents in Web transfers are heavy tailed [7]. In fact measurements on the Microsoft corporate WAN have shown that across other applications (such as Email, Transfers, Remote Desktop) transfer sizes are subexponential and well modelled by a Pareto or Lognormal distribution. The above results for resource pooling are insensitive to the distribution of the demand distribution, and so hold for these distributions provided the means are finite.

This is for a static, known demand, whereas in practice this is at best estimated, and the capacities may also be unknown to the sender. Suppose that the overall demand 2ν is known, and that it is split amongst the two resources in proportion p and $1-p$. Then the average response time is

$$F\left(\frac{p}{C-2\rho p}+\frac{1-p}{C-2\rho(1-p)}\right) \tag{2}$$

which is strictly larger than equation (1) unless $p = 0.5$, illustrating the effects of uncertainty. Moreover the system will be unstable unless

$$C > 2\rho \max(p, 1-p)$$

whereas the combined system is stable provided $C > \rho$. In other words, resource pooling gives better performance and a larger feasible region. If the two resources represent alternate paths, or dual-homed paths, this illustrates the benefits of joint routing or dynamic routing rather than static routing.

We could similarly explore the cases where the resources have different capacities, $C_1 > C_2$ say, showing how uncertainties in capacities when used with static of fixed routing also translate into poorer performance. On a longer timescale, this can model the effects of failures, where capacity on a path becomes reduced.

More generally, suppose we have a network represented by a capacitated graph $G = [X, J]$ where X is the set of nodes, J the set of edges (arcs) connecting the nodes represent. Here the edges represent resources, having capacities C_j, with demands of type r, $r \in \mathcal{R}$ for some countable set $\mathcal{R}$, each associated with some source-destination pair i, j, for some $i \in X, j \in X$. We now associate a set of resources (edges) with a route in the network, where a route is a connected subgraph, and there is a set $\mathcal{S}$ of of routes $s \in \mathcal{S}$ available in the network where now we let $A = (A_{js})$ denote the link-route incidence matrix. Each type-r flow may split its traffic among a subset of routes. Let B be the route-flow incidence matrix, with $B_{sr} = 1$ if type r-flows may use route s, and $B_{sr} = 0$ otherwise.

We characterise the demand for each type r flow by a Poisson process with rate ν_r and exponentially distributed demand size with mean μ_r^{-1}, with offered load $\rho_r = \nu_r/\mu_r$. Now suppose that we have a performance measure associated with a single resource of capacity C of the form $\Gamma(\rho, C)$, that depends on the demand only through the offered load , with Γ a real valued function that is non-decreasing in its first argument and non-increasing in the second argument, with the natural ordering operator $\leq$. For example $\Gamma(\rho, C) = 1/(C-\rho)$ captures mean response time, $\Gamma(\rho, C) = \rho - C$ relates to stability, and we might have more general functions that capture distributions or their statistics. Dimensioning the network entails ensuring adequate performance, which can be translated to be $\Gamma(\rho, C) \in \mathcal{D}$, for some measurable set $\mathcal{D}$; for example for the stability function Γ given above we have $\mathcal{D} = (-\infty, 0)$.

If the demand is offered to a path which is a set of resources, C_r *each* of which are required to serve the demand, then the performance measure is $\cap\Gamma(\rho, C_r)$ where $\cap$ is a generalised

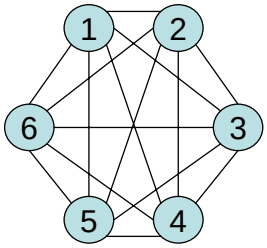

Fig. 1. A full mesh topology

conjunction operator (such as '+' or 'max' depending on the algebra), and since such additional resources ('in series') can only degrade performance it is natural to require

$$\Gamma(\rho, c_e) \leq \cap\Gamma(\rho, C_r) \forall c_e \in C_r. \tag{3}$$

In contrast, if there is as set of resources C_r some or all of which may be used together to serve the demand, then the performance $\oplus\Gamma(\rho, C_r)$ naturally satisfies (with a slight abuse of notation)

$$\oplus\Gamma(\rho, C_r) = \Gamma(\rho, \oplus C_r) \leq \Gamma(\rho, c_e) \forall c_e \in C_r. \tag{4}$$

where $\oplus$ is a generalised addition operator. If we have true resource pooling, then we naturally $\oplus$ identified with $+$, or a generalisation of it. We can similarly handle aggregating demands; for now we shall assume that demands are additive with the natural operator applied to loads.

Now consider a cutset of the graph G (recall a cutset disconnects the graph and no proper subset of it would disconnect the graph). Let $\mathcal{C}$ be such a set. Then under resource pooling, a necessary condition for the performance to be met is that

$$\Gamma(\sum_{\cup_{j\in\mathcal{C}} r:c_j\in r} \rho_r, \sum_{j\in\mathcal{C}} C_j) \in \mathcal{D} \tag{5}$$

where $r : c_j \in r$ is shorthand for $r : A_{js}B_{sr} = 1$ for some $s \in \mathcal{S}$. In other words, we sum the demand over the demands of type r for which the source or destination of r is one of the two components of G caused by removing the cutset.

The number of cutsets grows exponentially with the graph size, and jointly satisfying conditions (5) for a set of cutsets is necessary but not necessarily sufficient for meeting the performance objective. However, in many cases we can consider relatively simple cutsets, and dimension to these to meet overall objectives, namely, the conditions become sufficient. Cutset dimensioning is attractive when it works: it says we only need to be able to forecast (measure) aggregate traffic, and says that provided there is enough capacity, it is not critical where that capacity is, provided it can be found by the routing scheme.

A. Mesh topology

One example where cutsets have proved effective is the full mesh topology illustrated in Figure 1. There are N nodes, and a directional link with capacity C between any ordered node pair.

This topology has been considered in [8], [9] as a candidate backbone architecture and has been used extensively in telephony networks [10]. The authors in [8] consider a so-called Valiant Load Balancing (VLB) which split the traffic demand from node i to node j is 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 [8] show that for any traffic matrix $(\rho_{ij})_{i,j\leq N}$ 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, which is equivalent to choosing node cutsets: namely those cutsets formed by removing all incoming or outgoing edges from a node.

In our multi-path routing scheme, data is transferred from node i to node j along two simultaneous paths taken from the set of direct or two-hop paths, with continuous re-sampling to identify efficient paths. Such a scheme satisfies the cutset constraints and hence is feasible. It can handle any traffic that VLB can handle and also cope with demands that cannot be handled by VLB. For instance 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 $2(1-1/N)$.

We have been deliberately vague about the particular stochastic model used, since this approach can apply equally well to loss networks, fast-packet networks or traditional packet networks. Informally, the potential gains are much greater for delay/queuing networks than loss networks, since the multiplexing gains are higher[2]. There is an issue with a traditional store-and-forward based queueing network: for such open-queuing networks, [11] showed that even for stability it is necessary to consider *generalised cutsets* as resource pools in order to obtain necessary and sufficient conditions for stability: they correspond to linear combinations of resources and the corresponding loads but are not cutsets. The Internet behaves as a packet network or a fast-packet network depending on the timescale we consider; with buffer sizes shrinking in relative terms, it becomes even more appropriate to consider flow-level dynamics.

III. Multi-path routing

We now look at how multipath routing and congestion control might work in more detail. We perform load balancing at two levels. At the lower level, a 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 therefore different from the load-balancing algorithms of both [8] and [12] which split load to paths

[2]for a loss network, eg using $M/M/C/C$ the benefits of aggregating, such as putting two such systems together, decrease as C increases

according to fixed splitting ratios, irrespective of changes in traffic conditions. It operates on a time scale dictated by the RTTs of the current paths. At the higher level, we periodically resample for new paths, which is done via random selection of paths. 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 [13]. 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 $x = \frac{1}{RTT\sqrt{p}}$.

A. Rate optimisation

We now describe multipath routing, building on the stochastic model outlined in Section II. We associate a utility function U_r with calls of type r, which we assume is an increasing, strictly concave function on $\mathbb{R}_+$, and continuously differentiable. For convenience, we further assume that $U_r'(x) \to \infty$ as $x \downarrow 0$ to force non-zero rate allocations, and that $U_r'(x) \to 0$ as $x \uparrow \infty$. In addition we redefine the notion of a performance measure $\Gamma(\cdot,\cdot)$ to be a penalty function, or "cost" Γ_j associated with each resource $j \in J$, assumed to be convex and non-decreasing in its first argument (and non-increasing in its second).

In addition to convexity, we shall make the following critical assumptions about the penalty functions $\Gamma_j(z;C)$.

$$L\Gamma(z;c) \equiv \Gamma(Lz;Lc), \quad z \in \mathbb{R}_+,\ c \in \mathbb{R}_+,\ L > 0. \tag{6}$$

This condition ensures that the rate allocation x is left unchanged after simultaneously rescaling by some number L both the numbers of flows and the capacities, which we need later.

These conditions are naturally satisfied when we can write $\Gamma_j(y_j;C_j) = \int_0^{y_j} p_j(\eta/C_j)d\eta$ as the rate at which "cost" is incurred at the resource, and where we can interpret $p_j(y_j/C_j)$ as the probability of dropping (or marking) a packet at resource j when the load on the resource is y_j and its capacity is C_j. Such models arise naturally when equating resources with output ports on routers, which have limited buffering. For example setting $p_j(y_j/C_j) = [y_j - C_j]^+/[y_j]$ corresponds to treating the resources as bufferless resources, while $p_j(y_j/C_j) = \min(1,(y_j/C_j)^b)$ models the case of small buffers (meaning $b = o(C)$) where packets are dropped or marked when the buffer contents exceed b.

Suppose for the moment that we fix the number of flows of type r at N_r, then the (social) optimum rate allocations x_r solve the optimisation problem

$$\begin{aligned} \text{Maximise} \quad & \sum_{r\in\mathcal{R}} N_r U_r\left(\sum_{s\in\mathcal{S}} B_{sr}x_{sr}\right) \\ & -\sum_{j\in J}\Gamma_j\left(\sum_{r\in\mathcal{R}} N_r \sum_{s\in\mathcal{S}} A_{js}B_{sr}x_{sr}; C_j\right) \\ \text{over} \quad & x_{sr} \geq 0,\ r\in\mathcal{R},\ s\in\mathcal{S}. \end{aligned} \tag{7}$$

The variable x_{sr} represents the sending rate of type r-users along route s, and $\Gamma_j(y_j;C_j)$ is the cost for sending at rate y_j over link j, assumed to be convex and non-decreasing. The per-user rates are $x_r = \sum_{s\in\mathcal{S}} B_{sr}x_{sr}$.

The multipath formulation was first described in [13].

Recent research [14], [2], [1] has shown it is possible to design efficient multipath controllers that rely only on local path information and which perform this optimisation implicitly in a distributed fashion. Such controllers are TCP-like: for each path there is a steady increase of the rate and 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.

Note the *single* utility function for type r traffic, which allows coordination across routes. Without such coordination, we can both lose efficiency and incur higher costs [17]. 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 remove the RTT bias altogether.

This form of optimisation is a natural one, allowing general cost functions. Recently Kandula et al. [15] have looked an intra-domain optimisation which looks at minimising the maximum utilisation.

B. Fluid model dynamics

Recall the dynamics of elastic traffic, outlined in Section II to which we now add the dynamics for the streaming traffic and specify the bandwidth sharing process. We assume streaming traffic of type r arrives as a Poisson process of rate κ_r, has an exponentially distributed holding time with mean η_r^{-1}, and receives a rate allocation x_r, which is a solution to equation (7) . Note that x_r is a function of the number of type r calls in progress in the network, and x_r is rate allocated to flows of type r, irrespective of whether they are elastic or streaming. For convenience, we write M_r instead of N_r in the above if the flow is a streaming flow, and write $(\mathbf{N},\mathbf{M}) = (N_r, r\in R; M_r, r\in R)$. Given our assumptions, $(\mathbf{N},\mathbf{M})$ is a Markov process on $\mathbb{Z}_+^{\mathcal{R}} \times \mathbb{Z}_+^{\mathcal{R}}$ For example considering file tranfsers, transisition rates from $(\mathbf{N},\mathbf{M})$ to $(\mathbf{N}+e_r,\mathbf{M})$ occur at rate ν_r, and transitions to $(\mathbf{N}-e_r,\mathbf{M})$ occur at rate $\mu_r N_r x_r$, where e_r is the unit vector with 1 in position r.

We can now consider a fluid limit by scaling the number of flows by a number L, and considering $(\mathbf{n},\mathbf{m})(t) = (\mathbf{N}_L(t)/L, \mathbf{M}_L(t)/L)$ as $L \to \infty$, where $(\mathbf{N}_L(t),\mathbf{M}_L(t))$

is the above model but with $C_j, j \in J$, and $\nu_r, \kappa_r, r \in \mathcal{R}$, replaced by $LC_j, j \in J$, and $L\nu_r, L\kappa_r, r \in \mathcal{R}$, respectively. Is them possible to show [16] that the system coverges to the set of differential equations

$$\begin{aligned} \frac{d}{dt} n_r(t) &= \nu_r - \mu_r n_r(t) x_r(\mathbf{n}(t) + \mathbf{m}(t); \mathbf{C}). \quad r \in \mathcal{R} \\ \frac{d}{dt} m_r(t) &= \kappa_r - \eta_r m_r(t), \quad r \in \mathcal{R}. \end{aligned} \tag{8}$$

C. Limits, stability and optimality

Theorem 3.1: Under multipath routing, the differential equations (8) have a unique invariant point, $(\hat{\mathbf{n}}, \hat{\mathbf{m}})$, that takes the form

$$\hat{m}_r = \kappa_r/\eta_r, \quad \hat{n}_r = \rho_r/\hat{x}_r, \quad r \in \mathcal{R}, \tag{9}$$

The system is stable in the sense of Lyapunov provided that Γ_j is strictly increasing, and satisfies mild regularity conditions, which ensure the system is not overloaded. The allocation of type r flows for given r to routes is such that at equilibrium non-zero allocations only occur on those routes s for which the "prices" $\sum_{j \in J} A_{js} \Gamma_j'(\hat{y}_j; C_j)$ are equal. When no streaming traffic is present, the offered load is split optimally across routes *independently* of the choice of utility functions U_r.

Proof: That the stated values are an equilibrium point follows by setting the derivates to zero in (8). Uniqueness and stability follow from the general results of [17], subject to conditions on U_r' and Γ' which essentially generalise natural stability conditions, and which in our case hold under our assumptions on U provided Γ is well-behaved. Now it follows from equation (7) that if at the equilibrium point $B_{sr} > 0$ then

$$U_r'(\sum_{s' \in \mathcal{S}} B_{s'r} \hat{x}_{s'r}) = \sum_{j \in J} A_{js} \Gamma_j'(\hat{y}_j; C_j) - \beta_{sr} \tag{10}$$

where $\hat{y}_j = \sum_{s \in \mathcal{S}, r \in \mathcal{R}} A_{js} B_{sr} (\hat{n}_r + \hat{m}_r) \hat{x}_{sr}$, and β_{sr} is the Lagrange multiplier associated with the constraint $x_{sr} \geq 0$ and satisfies the constraint qualification conditions $\beta_{rs} \geq 0$, $\beta_{rs} \hat{x}_{sr} = 0$. Γ_j' denotes the derivative of Γ_j. For any fixed r, it then follows that there exists a critical value p_r such that the "prices" $\sum_{j \in J} A_{js} \Gamma_j'(\hat{y}_j; C_j)$ on any route s such that $B_{sr} = 1$ must coincide with p_r if $\hat{x}_{sr} > 0$, and be less than p_r otherwise. Hence a non-zero allocation is only possible on equally priced routes.

Denote by ρ_{sr} the fraction $\hat{n}_r \hat{x}_{sr}$ of load ρ_r offered by type r-flows that, in equilibrium, is carried along route s. The above property justifies the following interpretation: with multipath routing, in equilibrium the load fractions ρ_{sr} are such that the overall cost

$$\sum_{j \in J} \Gamma_j \left(\sum_{s \in \mathcal{S}, r \in \mathcal{R}} A_{js} B_{sr} (\rho_{sr} + (\kappa_r/\eta_r) \hat{x}_{sr}); C_j \right)$$

is minimised. When no streaming traffic is present, this is exactly the solution we would obtain if we want to chose ρ_{sr} such that $\rho_r = \sum_s \rho_{sr}$ and wanted to minimise $\sum_{j \in J} \Gamma_j \left(\sum_{s \in \mathcal{S}, r \in \mathcal{R}} A_{js} B_{sr} \rho_{sr}; C_j \right)$, which is independent of U_r. ■

A few remarks are in order.

1) Unless "prices" are equal on different routes, only one route will be used.
2) In the absence of streaming traffic, the allocation of traffic fractions to routes is independent of the utility function (and so of the flow control algorithm used), however the performance does depend the utility function (through the allocation x_r).

IV. Route choices

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, 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 path 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. [18].

Informally, we can appeal to some the results related to DAR [3] or Mitzenmacher's work [4] to show in specific cases, such an approach does as well as if we truly spread load out as in the solution to the optimisation theorem.

Note that the solution to the optimisation only requires that we spread the load across the lowest cost routes in the right proportions, and any mechanism which achieves this has the same performance.

In fact under certain restrictions on cost functions, we can prove the following:

Theorem 4.1: Assume that class r-transfers can use any network paths from an associated set $\mathcal{P}_r$. 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 routing 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.

V. Architectural Issues

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. However, 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 [19] suggest that 60% of stub-ASes (those which do not transit traffic) are multihomed, and [20] claims that with IPv6 type multihoming there are at least two disjoint paths between such stub-ASes. Multihoming requires several addresses per end-system, which is made possible by IPv6.

Multihoming goes some way towards addressing the critical issues of diversity: for both performance and reliability reasons we would like at least 2 disjoint paths between source and destination. See also [12] 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 [12] 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.

VI. Concluding remarks

We have outlined the benefits of combining multi-path routing with coordinated congestion control, which can provide robustness, performance benefits and resilience through the use of resource pooling and efficient routing.

Potential application scenarios are for intra-domain routing, within a peer-to-peer cloud, and even the Internet itself, thereby giving a degree or routing autonomy to the end-systems.

We have only briefly commented on choice of routes and when to adapt: this is a subject of ongoing study, and the question of delayed feedback and interacting control loops warrants further investigation.

References

[1] F. P. Kelly and T. Voice, "Stability of end-to-end algorithms for joint routing and rate control," *Computer Communication Review*, vol. 35, no. 2, pp. 5–12, 2005.

[2] H. Han, S. Shakkottai, C. Hollot, R. Srikant, and D. Towsley, "Overlay TCP for multi-path routing and congestion control," 2004, submitted for publication to IEEE/ACM Trans. Networking.

[3] R. J. Gibbens, F. P. Kelly, and P. B. Key, "Dynamic Alternative Routing: modelling and behaviour," in *Teletraffic Science, Proceedings 12th International Teletraffic Congress*, M. Bonatti, Ed. Elsevier, Amsterdam, 1989, pp. 1019–1025.

[4] M. Mitzenmacher, A. Richa, and R. Sitaraman, *The power of two random choices: A survey of the techniques and results*. Kluwer, 2000.

[5] L. Kleinrock, *Queueing Systems*. Wiley, 1975/6, vol. 1-2.

[6] S. Floyd and V. Paxson, "Difficulties in simulating the Internet," *IEEE/ACM Trans. Networking*, vol. 9, no. 4, pp. 393–403, 2001.

[7] M. E. Crovella and A. Bestavros, "Self-similarity in World Wide Web traffic: Evidence and possible causes," *IEEE/ACM Trans. Networking*, vol. 5, no. 6, pp. 835–846, December 1997.

[8] R. Zhang-Shen and N. McKeown, "Designing a predictable Internet backbone network," in *HotNets*, 2004.

[9] N. M. R. Zhang-Shen, "Designing a predictable Internet backbone network," in *IWQoS*, 2005.

[10] R. J. Gibbens and F. P. Kelly, "Network programming methods for loss networks," *IEEE J. Select. Areas Commun.*, vol. 13, pp. 1189–1198, 1995.

[11] F. Kelly and C. Laws, "Dynamic routing in open queueing networks: Brownian models, cut constraints and resource pooling," *Queueing Systems*, vol. 13, pp. 47–86, 1993.

[12] K. Gummadi, H. Madhyastha, S. Gribble, H. Levy, and D. Wetherall, "Improving the reliability of Internet paths with one-hop source routing," in *OSDI*, no. 6, 2004.

[13] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, pp. 237–252, 1998.

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

[15] S. Kandula, D. Katabi, B. Davie, and A. Charny, "Walking the tightrope: Responsive yet stable traffic engineering," in *Proc. SIGCOMM 2005*, vol. 35, no. 4, CCR, August 2005, pp. 253–264.

[16] S. Kumar and L. Massoulié, "Fluid and diffusion approximations of an integrated traffic model," Microsoft Research, Technical Report MSR-TR-2005-160, 2005, http://research.microsoft.com/users/lmassoul/MSR-TR-2005-160.ps.

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

[18] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Rao, "Improving Web availiability for clients with MONET," in *NSDI*, 2005.

[19] S. Agarwal, C. Chuah, and R. Katz, "OPCA: Robust interdomain policy routing and traffic control," in *IEEE Openarch*. New York, NY, April 2003.

[20] C. de Lanois, B. Quoitin, and O. Bonaventure, "Leveraging internet path diversity and network performances with IPv6 multihoming," Universite catholique de Louvain, Technical Report 2004-06, 2004.