A Mutualistic Resource Pooling Architecture

João Taveira Araújo [*]
University College London
j.araujo@ee.ucl.ac.uk

Miguel Rio
University College London
m.rio@ee.ucl.ac.uk

George Pavlou
University College London
g.pavlou@ee.ucl.ac.uk

ABSTRACT

Parallelism pervades the Internet, yet efficiently pooling this increasing path diversity has remained elusive. We defend that the inability to progress beyond a single path paradigm is due to an inflexible resource sharing model, rather than a lack of routing solutions. The tussle between networks and hosts over resource sharing has constricted resource pooling into being redefined by stakeholders according to their own needs, often at the expense of others.

In this paper we debate existing approaches to resource pooling and present PREFLEX, an architecture where edge networks and hosts both share the burden and reap the rewards of balancing traffic over multiple paths. Using PREF (Path RE-Feedback), networks suggest outbound paths to hosts, who in turn use LEX (Loss Exposure) to signal transport layer semantics such as loss and flow start to the underlying network. By making apparent network preferences and transport expectations, PREFLEX provides a mutualistic framework where congestion control and traffic engineering can both coexist and evolve independently.

1. INTRODUCTION

While the Internet has become evermore interconnected, exploring path diversity has been relegated to an afterthought in an architecture modeled around assumptions that no longer stand. Single-path forwarding as a paradigm arose not as a guiding principle, but as a natural aversion towards increasing both the complexity and cost of a resource starved network.

Engineering for scarcity has propelled the Internet to an unprecedented scale, but problems arise when what was otherwise scarce becomes plentiful. Protocols designed to be bit conservative at the expense of latency have become technological anachronisms as bandwidth costs continue to plummet. Similarly, the notion of a router as a device merely capable of forwarding packets has long been obsolete as Moore's law continues to pave the way for greater functionality within the network. Network address translation (NAT), deep packet inspection (DPI) or performance enhancing proxies (PEP) are all examples that when it comes to drawing a boundary between network and transport, the line begins to blur [1].

Furthermore, parallelism seems to be a dominant trend at every level of the Internet architecture as a cost-effective means of increasing both performance and robustness. At the inter-domain level, the AS graph is becoming flatter and more highly interconnected [2]. Within domains, the sheer complexity of managing paths has led to the streamlined design and deployment of MPLS [3], implementing a fully fledged layer in its own right. At the edges the rise in multi-homing continues to increase the strain on an already overloaded routing architecture. Even within network components, parallelism is such that packet re-ordering can no longer be considered pathological [4].

Given these trends, one would expect the ability to pool traffic across such emergent path diversity to have become a network primitive. In reality, each stakeholder in the Internet architecture seems to balance traffic according to their needs while attempting to remain inconspicuous to others. At best, this interaction between stakeholders can be seen as a form of commensalism, where one entity can extract benefits while others remain unaffected. At worse, the competitive nature of the tussle [5] that ensues can spiral into a situation where few profit.

In this paper we investigate the nature of this antagonism between network and endpoints and reflect on how the Internet can accommodate the needs of both. We then introduce PREFLEX, Path RE-Feedback with

[*] The author is funded by the Portuguese government through FCT grant SFRH/BD/37855/2007, and would also like to acknowledge INESC Porto, where this paper was partially written.

ACM ReArch 2010, November 30, 2010, Philadelphia, USA.
Copyright 2010 ACM 978-1-4503-0469-6/10/11 ...$10.00.

Loss EXposure, an architecture for balancing congestion which foments mutualism between end-hosts and edge network providers.

2. A HISTORY OF ANTAGONISM

The ability to evolve beyond single path forwarding has often been misdiagnosed primarily as a routing challenge. The subject is frequently revisited with varying approaches [6, 7, 8, 9]. Despite this, multipath routing has remained a pipe dream for end-hosts. The common trait all these proposals share is a failure to identify the tussle over resource control as the primary obstacle in moving towards the use of multiple concurrent paths.

The Internet architecture places resource control at the edges, in what can be viewed as an instance of the end-to-end principle [10]. This represented a fundamental paradigm shift, ultimately conferring the scalability which fueled the growth of the Internet. While unilateral control of a network resource by hosts was already polemic in an academic research network, with the rise of the commercial Internet this notion has slowly been set aside by stakeholders intent on exerting control over their own networks.

Network operators have now become accustomed to inspect, shape and throttle traffic in an attempt to override resource sharing as implicitly performed by TCP. A common cause for such behaviour could derive from the perceived freeriding made possible by TCP, whereby a minority of users can gain an disproportionate amount of bandwidth, with detrimental effects for the majority of users. In a broader sense, networks attempt to reflect their own objectives and concerns. Because this was not contemplated when designing our resource sharing model the subsequent violations of the end-to-end principle say more about the limitations of the current architecture than the ill intent of the perpetrators.

Nowhere is this more apparent than in traffic engineering. Network operators rely heavily on traffic engineering to balance utilisation over long timescales in an attempt to reduce costs by making efficient use of available paths. Since information at the network layer is limited, traffic engineering optimizes for the wrong metric - utilization - in detriment of the congestion it may be causing. Additionally, this optimization is typically executed offline, and re-computed over long timescales to minimize the impact to higher layers and ensure stability. The limiting assumption is that traffic patterns are exogenous. In reality, hosts will often find means of adapting to network conditions, such as establishing overlay networks. This resulting shift in behaviour may in turn conflict with the concurrent traffic engineering process, which will have to readjust to a substantially different traffic matrix in a next iteration. This antagonistic cycle leaves traffic engineering as a whole stuck in a rut, unable to adapt too often, out of fear of disrupting transport protocols, and unable to adapt often enough in order to react to changes in traffic.

The inability to reach a compromise between network interests and transport layer expectations has severely limited network-assisted traffic balancing despite strong commercial interest. Equal-Cost Multipath (ECMP), a seemingly simple solution for balancing traffic over similar paths, has fallen out of grace [11]. Relevant research in dynamic traffic engineering, such as MATE [12] or TeXCP [13], has trod new footsteps on old grounds, continuing to focus on utilisation as the sole metric for performance.

In stark contrast with traffic engineering, the interest in the use of congestion control to balance traffic across paths has gained significant traction, particularly in the wake of seminal contributions [14], [15] which provide the theoretical basis for much of the standardization effort behind Multipath TCP [16] (MPTCP). This alone however is unlikely to overcome significant architectural shortcomings. For one, path diversity is opaque to end-hosts, which restricts deployment of MPTCP to multihomed hosts. Given networks are already concerned with TCP's ability to share bandwidth in its single path incarnation, it remains unlikely ISPs will consider making path diversity visible to end-hosts. Additionally, it is not yet clear what proportion of traffic MPTCP will encompass, or what proportion of MPTCP traffic would be required to maintain a network consistently balanced. Finally, flows which cannot be split into subflows, because they are too short or for other motives such as security concerns, will remain restricted to choosing a default path, rather than a best path.

Neither congestion control or traffic engineering alone seem fully capable of bridging the divide between networks and end-hosts. The discussion around the relative merits of both is often manichaean and erroneously simplified as a conflict between advocates and opposers of the end-to-end principle. This entirely misses the point. The concern should not revolve around whether an approach is right or wrong, but whether it is applicable within a given context or not. The recognition of the commercial network as a fundamental stakeholder is intrinsic to the evolvability of the current architecture. In the absence of such recognition, the gulf between our perception of how the Internet works and how it works in practice will only widen. If we regard both traffic engineering and congestion control as different sides of the same coin, it is our duty to provide the architectural underpinnings for both to evolve independently while not foregoing cooperation.

Recent research in resource sharing has suggested that much of the misalignment between network and transport derives from the lack of accountability for congestion. While previous work had modelled and analyzed the broken incentive structure subjacent to forwarding traffic from an economic perspective, work on re-feedback and congestion exposure [17] pioneered a practical means of alleviating the tussle surrounding resource sharing. In particular, congestion exposure advocates the use of congestion volume, rather than throughput or traffic volume, as the by-product by which the impact of traffic should be assessed. We build on this approach and present a concept for a joint, mutualistic architecture for congestion control and traffic engineering.

3. PREFLEX ARCHITECTURE

The PREFLEX (Path RE-Feedback with Loss EXposure) architecture can be split into two independent components. At the network, we define a mechanism for path re-feedback (PREF), whereby stub domains can signal a preferred path to end-hosts according to local policy or perceived path quality. At the end-hosts we specify a transport agnostic protocol for loss exposure (LEX), which explicitly marks packets within a flow in order to signal path loss back to the network.

While functionally separate, in practice both components work in tandem. The use of loss exposure, while executed by hosts, provides network operators with feedback on end-to-end path loss. Conversely, with path re-feedback hosts are allowed access to paths selected by the network. Together, PREFLEX bridges the divide between network and transport layers in order to balance congestion, rather than load, over the multiple paths typically available solely to edge networks.

3.1 Loss Exposure

We propose a simple protocol for revealing loss, LEX, which not only borrows heavily from re-ECN [17], a protocol for congestion exposure, but which can coexist and serve as a stepping stone for the deployment of the latter. By revealing information currently confined to the transport layer down to the network, we are both reducing the need for the network to inspect higher level protocol headers in order to redistribute bandwidth differently and correcting the information asymmetry that currently afflicts networks, who know less about the quality of service they provide than their customers.

3.1.1 From flow to flowlet

The first change proposed for LEX is to have end-hosts mark, at the network layer, packets belonging to flows where feedback has not been established. This typically corresponds to the first packet exchange in a flow, such as SYN packets in TCP, but may also include the first packet after a significant idle period, a keep-alive packet or a renewed attempt at a retransmission after successive timeouts in the case of network failure. Within LEX, as with re-ECN, such packets are labelled FNE (Feedback Not Established).

The signalling of such packets has many practical implications. For one, from simply inspecting the IP header, networks are made aware of the first of a succession of similar packets, which poses significant advantages in allocating state in middleboxes, whether it be to perform admission control, policing or traffic shaping. All of the above are possible by inspecting TCP, but we attempt here to make apparent an architectural illusion: that a connectionless layer should be oblivious to connection setup. By making such information explicit at the IP layer we are alleviating in some measure the need for consistent violation of layering by network equipment, or hopefully circumscribing such practices to a small subset of packets.

Additionally, the concept of a transport flow, which establishes an association between two endpoints, is decoupled from the concept of a network flow, which will henceforth be referred to as a flowlet [18]. We define a flowlet as a stream of packets which the endhost expects to follow the same network path. The same transport flow may be composed of a single flowlet, parallel flowlets, or a succession of different flowlets. As we shall see later, this feature is particularly advantageous for balancing traffic as flowlets provide a finer granularity than existing flows, as well as allowing flows to quickly switch path without breaking the transport session.

3.1.2 Echoing loss

Once feedback has been established, hosts adjust their sending rate in response to implicit congestive signals such as delay or packet loss, or explicit signals such as ECN. Protocols for congestion exposure, such as re-ECN, mark outgoing packets according to the explicit congestion marking received from the network. As such, IP packets would carry two congestion markings. The first indicating the congestion experienced so far and the second indicating the end-to-end congestion experienced by the host in the previous RTT. With this re-feedback of congestion markings, networks are able to estimate rest-of-path congestion, which is an important metric for keeping customers accountable for the congestion they cause and providers accountable for the services they offer.

We specify a simplified form of congestion exposure which uses the implicit information contained in losses as opposed to relying on the widespread deployment of congestion notification. Where packet loss does arise, LEX requires that hosts mark their respective retransmits with a Loss Experienced (LEx) codepoint. The drawback of this approach is that only the end-to-end congestion can be estimated from a stream of packets, which implies that traffic can only be reliably aggregated close to the source, and effectively policed close to the receiver. As we shall see later, because the focus

Codepoint	Meaning
Not-LECT	Not Loss Exposure Capable Transport
LECT	Loss Exposure Capable Transport
LEx	Loss Experienced
FNE	Feedback Not Established

Table 1: LEX codepoints and description.

of PREFLEX is balancing congestion at a stub domain this limitation is not significant.

If run as a complement of re-ECN, three of the four codepoints in table 1 are potentially shared, in which case only the loss experienced codepoint has to be added to the re-ECN specification. For routers along the path, an accurate estimate of the end-to-end path loss can be obtained by simply dividing the sum of bytes marked with the loss experienced codepoint, by the total traffic marked as either LECT or LEx. Additionally, one could envision a preferential dropping mechanism which prioritizes retransmits.

3.2 Path Re-feedback

For networks, the most significant hurdle in adopting multiple paths for a single destination has not been the selection process. Instead, the main difficulty resides in assigning packets to paths. Since balancing traffic at a packet granularity has severe repercussions for the transport layer, network operators have typically resorted to splitting prefixes. Increasingly, networks have also been able to afford the cost of keeping flow state in an attempt to balance traffic at a finer granularity.

Neither of these approaches are strictly necessary in a mutualistic architecture. Since we require that hosts be made aware of the path packets take, we can push flow state towards the edge by placing the responsibility for assigning packets to paths at the endpoints. In such a case, a network only needs to perform path selection according to local policy and pass the information onto the end-host.

For this purpose, we use FNE packets, as defined in LEX, to act as network triggers for path selection. An ISP or stub domain, upon detecting an incoming FNE packet, selects a preferred outgoing path based on the reverse lookup of the source address, and marks the packet with a path identifier. For IPv4, a possible location for such marking to occur could be within the Diffserv field, where a set of codepoints are reserved for local use. On receiving an FNE packet containing a path identifier, a sender should tag all subsequent packets in the flowlet using the same identifier in order to ensure it will traverse the selected egress at the edge domain.

A subtle implication of triggering path selection based on incoming packets, rather than resorting to out-of-band signalling for example, is that path selection becomes receiver driven. The responsibility for defining a strategy on when and how often to attempt a path request lays firmly with the stakeholder who extracts the most benefit. The flipside is that because FNE packets require additional network intervention, whether for selecting a new path or setting up state, networks may rate limit the amount of FNE packets they receive in order to protect themselves from overload. This is the current line of thinking with re-ECN, where FNE packets are used to set state in congestion policers.

4. BALANCING BY PREFLEX

PREFLEX establishes the mechanism by which a path is relayed back to an endpoint and the information an endpoint should provide the network with. The means by which a network selects a preferred path has been purposely left out thus far because it is clearly a run-time, rather than design, decision. Nonetheless, we will present a path selection algorithm which illustrates how PREFLEX widens the scope of traffic engineering.

Our balancer maintains individual routing tables associated to each domain egress. For each table entry we associate a flowlet ratio f_{di}, which defines the fraction of flowlets to a destination d that should be assigned to route i. Now consider only LEX enabled traffic destined to prefix d at a PREFLEX balancer.

Let T_i be the number of bytes sent through route i for the previous time period. Let L_i be the number of bytes marked with the loss experienced codepoint and sent through route i in the previous time period. Let N be the number of available routes for the given destination prefix. Let $L = \sum_i L_i$ and $T = \sum_i T_i$.

Splitting traffic may follow distinct approaches. One approach is to attempt to equalise utilisation. When small adjustments to traffic splits are preferred, there may be a desire to be conservative and maintain the existing traffic split. Finally, there may be the desire to balance losses. Call these splits $f(E)_i$, $f(C)_i$ and $f(L)_i$ where E, C and L stand for “equal”, “conservative” and “loss driven”. Use the dash notation for the same quantities in the next time period. Then our final distribution of traffic across all routes is:

$$f_i' = \beta_E f'(E)_i + \beta_C f'(C)_i + \beta_L f'(L)_i \qquad (1)$$

where $\beta_\bullet$ are user set parameters in $(0, 1)$ such that $\beta_E + \beta_C + \beta_L = 1$. Now by definition $f'(E)_i = 1/N$ and $f'(C)_i = T_i/T$, hence we need only define $f'(L)_i$. We wish to choose a distribution that will equalise the loss ratio $p_i = L_i'/T_i'$ for all permitted routes i. While the loss rate is an unknown function of T_i and bottleneck link bandwidth B_i, it is reasonable to assume that the loss rate is increasing with T_i and decreasing with B_i. Whatever the true function is, we can assume that in a small region around the current values of T_i and B_i,

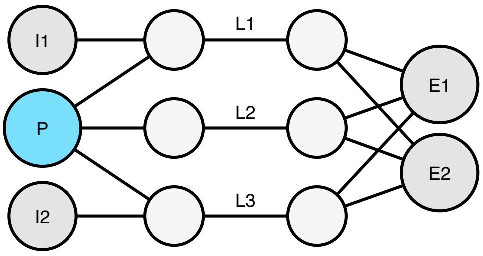

Figure 1: Simulation topology.

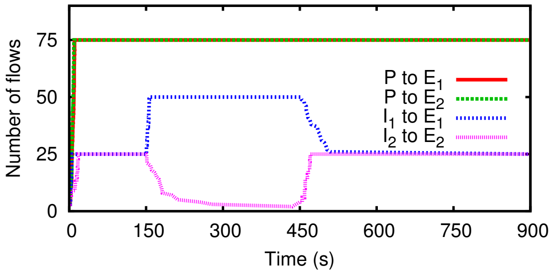

Figure 2: Number of concurrent flows set between ingress and egress domains.

loss is locally linear. In such cases, and assuming on average $T' = T$, we can deduce $f'(L)_i$ to be:

$$f'(L)_i = \frac{T'_i}{T} = \frac{T_i^2}{L_i \sum_i (T_i^2/L_i)} \tag{2}$$

We have not included the proof due to length constraints. Replacing (2) in (1), we can now write the complete function for calculating the ratio of flowlets to be assigned to a given route i:

$$f'_i = \beta_E \frac{1}{N} + \beta_C \frac{T_i}{T} + \beta_L \frac{T_i^2}{L_i \sum_i (T_i^2/L_i)} \tag{3}$$

Using this algorithm, we now illustrate how PREFLEX balances traffic with a simple simulation.

The chosen topology, shown in figure 1, presents ingress domains I_1,I_2 and P and egress domains E_1, E_2 interconnected by bottleneck links $L_1 = L_2 = L_3 = 25$Mbps. Clients connected to the egress domains request flows from sources connected to the ingress domains. Flow sizes are chosen randomly following a Weibull distribution with shape parameter $k = 0.5$ and scale parameter $\lambda = 10^6$. The number of concurrent flows at any given time is shown in figure 2. Neither clients nor sources are shown in figure 1. Each link has a $10ms$ delay, giving a total base RTT of $100ms$. Only domain P is PREFLEX aware, and balances traffic accordingly.

In figure 3 we compare loss ratios to destination E_1 using different strategies. We first set $\beta_E = 1$ to mimic traffic engineering and equalise the utilisation across all outgoing links, and then repeat the simulation using $\beta_E = 0.1$, $\beta_L = 0.9$ to produce an aggressive loss balancer.

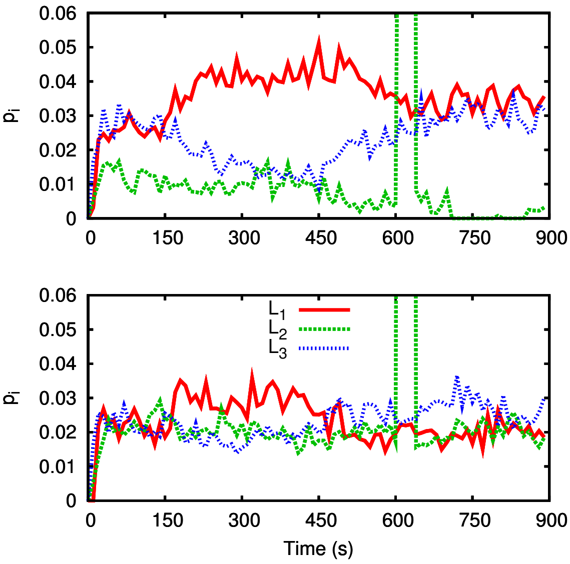

Figure 3: Loss ratio p_i for destination E_1 as seen by balancer P in equalisation mode (above) and loss balancing mode (below).

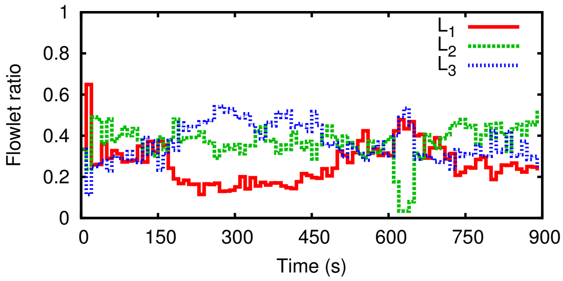

Figure 4: Flowlet ratio f_i for destination E_1 as set by balancer P in loss balancing mode.

Equalising utilisation alone is myopic, pushing traffic irrespective of congestion further upstream. Even when bottlenecks are of equal bandwidth, equalising utilisation can easily lead to substantially different loss rates. Furthermore, for lower volumes of flows splitting traffic at a flowlet, rather than packet, granularity can lead to uneven utilisation.

Loss balancing offers far more predictable performance for transport protocols while making upstream providers accountable for the quality of service they provide. It does so with no notion of the bottleneck bandwidth, flow length or number of flows already in the system. The update interval can be set arbitrarily small without conflicting with existing transport flows. In our simulation, the flowlet ratio was calculated every $10s$, as shown in figure 4. This allowed it to rapidly react to a link failure in domain E_1 at $600s$, which affected traffic routed through L_2. The flowlet ratio for the route passing L_2 quickly reached the minimum value β_E/N, set to ensure that paths continue to be probed even in the presence of high levels of loss. Even without TCP extensions for PREFLEX assisted path recovery, once link failure recovered at $630s$, both flowlet ratio and loss recovered in a short time. By comparison, equalising by utilisation failed to recognize an anomaly and routed a third of all new flowlets into the link failure.

We believe this form of smart routing which is able to dynamically adapt to existing network conditions is essential, both in providing robustness and fostering competition between providers. Furthermore, PREFLEX provides additional deployment incentives for ECN where it is most needed, at the edges, while maintaining the core accountable for losses that may occur, irrespective of the nature of congestion, such as the extreme case of network failures.

5. ACKNOWLEDGEMENTS

This work was partly funded by the EU FP7 COMET project - Content Mediator Architecture for Content-Aware Networks, grant agreement 248784.

6. CONCLUSIONS

We have broadly described an architecture which shares the responsibility for resource pooling between endhosts and edge networks, but does not explicitly dictate an outcome. PREFLEX has been designed to take into account the inevitable tussle which will occur between both, and we envisage use cases where control over resource pooling could feasibly shift entirely in one direction or the other.

At its most liberal, PREFLEX enables resource pooling to be entirely performed by end-hosts. At its most conservative, PREFLEX affords edge network providers more fine-grained control over traffic than ever before. Between either extreme, the resulting mutualistic architecture offers greater transparency, control and robustness by realigning the interface between network and transport in order to accommodate the needs of both.

7. REFERENCES

[1] B. Ford and J. Iyengar, “Breaking up the transport logjam,” *Proceedings of the 7th ACM Workshop on Hot Topics in Networks*, 2008.

[2] H. Haddadi, D. Fay, S. Uhlig, A. Moore, R. Mortier, and A. Jamakovic, “Mixing biases: Structural changes in the as topology evolution,” *Traffic Monitoring and Analysis*, pp. 32–45, 2010.

[3] E. Rosen, A. Viswanathan, and R. Callon, “Rfc3031: Multiprotocol label switching architecture,” *Internet RFCs*, 2001.

[4] J. Bennett, C. Partridge, and N. Shectman, “Packet reordering is not pathological network behavior,” *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 789–798, 1999.

[5] D. Clark, J. Wroclawski, K. Sollins, and R. Braden, “Tussle in cyberspace: defining tomorrow’s internet,” *IEEE/ACM Transactions on Networking*, vol. 13, Jun 2005.

[6] C. Sunshine, “Source routing in computer networks,” *ACM SIGCOMM Computer Communication Review*, vol. 7, no. 1, 1977.

[7] X. Yang, “NIRA: A new Internet routing architecture,” *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 4, 2003.

[8] X. Yang and D. Wetherall, “Source selectable path diversity via routing deflections,” *ACM SIGCOMM 2006*, Aug 2006.

[9] P. Godfrey, I. Ganichev, S. Shenker, and I. Stoica, “Pathlet routing,” *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, pp. 111–122, 2009.

[10] J. Saltzer, D. Reed, and D. Clark, “End-to-end arguments in system design,” *ACM Transactions on Computer Systems (TOCS)*, vol. 2, no. 4, p. 288, 1984.

[11] D. Thaler and C. Hopps, “Rfc2991: Multipath issues in unicast and multicast next-hop selection,” *RFC Editor United States*, 2000.

[12] A. Elwalid, C. Jin, S. Low, and I. Widjaja, “Mate: multipath adaptive traffic engineering,” *Computer Networks*, vol. 40, no. 6, pp. 695–709, 2002.

[13] S. Kandula, D. Katabi, B. Davie, and A. Charny, “Walking the tightrope: Responsive yet stable traffic engineering,” *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, 2005.

[14] P. Key, L. Massoulie, and P. Towsley, “Path selection and multipath congestion control,” *INFOCOM 2007. 26th IEEE International Conference on Computer Communications.*, pp. 143–151, 2007.

[15] F. Kelly and T. Voice, “Stability of end-to-end algorithms for joint routing and rate control,” *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 2, p. 12, 2005.

[16] D. Wischik, M. Handley, and M. Braun, “The resource pooling principle,” *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 5, pp. 47–52, 2008.

[17] B. Briscoe, A. Jacquet, C. D. Cairano-Gilfedder, A. Salvatori, A. Soppera, and M. Koyabe, “Policing congestion response in an internetwork using re-feedback,” *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, 2005.

[18] S. Sinha, S. Kandula, and D. Katabi, “Harnessing tcp’s burstiness with flowlet switching,” *Proceedings of the 3rd ACM Workshop on Hot Topics in Networks*, 2004.