Capacity bounds and robustness in multipath networks

Andrei Iu. Bejan*
Smith Institute, Surrey
Research Park, Guildford, UK
andrei.bejan
@cl.cam.ac.uk

Richard J. Gibbens
Computer Laboratory,
University of Cambridge, UK
richard.gibbens
@cl.cam.ac.uk

Robert Hancock
Roke Manor Research Ltd
Romsey, Hampshire, UK
robert.hancock
@roke.co.uk

Don Towsley
School of Computer Science
University of Massachusetts
Amherst MA 01003, USA
towsley@cs.umass.edu

ABSTRACT

The recent developments of multipath data transport protocols such as Multipath TCP allow end-systems to explore and share available resources within networks. Through dynamic load balancing over subflows these protocols ensure high levels of robustness to network failures and traffic overloads. In this paper we use fluid models to study the benefits that accrue when load is shared across subflows. We combine insights gained from the fluid models with a precise description of the capacity region for the network and show that our models of multipath protocols approach the boundary of the capacity region as the intensity of the offered traffic approaches a critical value. We quantify the extent to which multipath protocols will make a network robust to unforeseen traffic mismatches and link failures and illustrate our results with parameterised models for random fluctuations in the offered traffic.

Categories and Subject Descriptors

C.2 [**Computer-communication networks**]: Packet switching networks

Keywords

Mutlipath, capacity bounds, performance analysis

1. INTRODUCTION

Communications based on single end-to-end paths can easily expose data transmission to the risk of instability and disruption. In contrast, multipath extensions of data transmission protocols aim to take advantage of path diversity in order to achieve efficient bandwidth allocation while maintaining stability and connectivity. The recent developments of multipath data transport protocols such as Multipath TCP (MPTCP) allow multi-homed end-systems with potential access to a rich set of technologies and paths (e.g. 3G, 4G, 802.11, satellite) to explore and share available resources by using existing path diversity within networks [19]. Through dynamic load balancing over subflows that can harness the available path diversity, these protocols have the potential to ensure certain levels of robustness to network failures and traffic overloads. Such multipath resource pooling extensions of data routing and congestion control implement decentralisation with implicit resource sharing and, at the same time, may advantageously implement coordinated control where the rates over available paths are determined as a function of all or some of the available paths [13].

During the last decade researchers have studied various properties and performance issues related to design and deployment of MPTCP [18]. Wischik et al (2009) [25] showed how the path choice offered by the network affects the ability of end-systems to shift their traffic across a pool of resources. They defined and studied a *resource poolability* metric, which measures for each resource how easy it is for traffic to be shifted away from that resource, e.g. in the event of a traffic surge or link failure. Chen et al [4] provided a thorough field study and explored the performance of MPTCP in the wild focusing on two path scenarios and investigating how much benefit can arise from using multipath TCP over cellular and WiFi relative to using only a single interface alone. Wischik et al (2011) [26] and Khalili et al (2012) [14] considered various fairness properties as well as other aspects of MPTCP. Link utilisation and route flappiness issues of multipath controllers were analysed in [9]. Raiciu et al (2011) [22] proposed to use completely distributed and topology agnostic MPTCP as a replacement for TCP in data centers, as it more effectively and seamlessly uses available bandwidth, giving improved throughput and better fairness on many topologies. These authors conclude that MPTCP is highly capable of finding free capacity in the network and increasing fairness, and is robust to congested links or failures because it combines path selection, scheduling and congestion control.

*Work undertaken while at the Computer Laboratory, University of Cambridge.

VALUETOOLS 2015, December 14-16, Berlin, Germany

DOI 10.4108/eai.14-12-2015.2262625

Design and stability of multipath end-to-end congestion and rate/routing control algorithms can be studied and conveniently formulated at the scale of flow rates. In recent years researchers have developed a framework that allows congestion control algorithms to be interpreted as distributed mechanisms solving a global optimisation problem [10,20,21, 23,24]. The framework is based on fluid-flow models, and the form of the optimisation problem makes explicit the equilibrium resource allocation policy which can often be restated in terms of a fairness criterion.

In this paper we use fluid models of scalable MPTCP to study the benefits that accrue when load is shared across subflows. We first obtain a precise description of the capacity region of a communication system with multipath control and then combine this description with insights gained from the fluid models showing that our models of multipath protocols can approach the boundary of the capacity region as the intensity of the offered traffic approaches the critical value. We consider the extent to which multipath protocols will make a network robust to unforeseen traffic mismatches and illustrate our results with parameterised models for random fluctuations in the offered traffic.

2. NETWORK MODEL

2.1 Network structure

We assume now that the network consists of a collection of *source-destination pairs* S and a collection of *links* J. These links represent capacitated directed edges between adjacent forwarding nodes. Each $s \in S$ identifies a unique source-destination pair (we shall therefore also refer to s as *source*) which is also associated with a set of *routes* R. Each such route $r \in R$ is a set of links, that is $r \subseteq J$. If source s transfers data over route r, then we write $r \in s$. Similarly, if a route r uses link $j \in J$, we write $j \in r$. Furthermore, $s(r) \in S$ refers to the unique source/destination pair identified with the route r.

Let T_{rj} be the propagation delay from source $s(r)$ to link $j \in r$, that is the length of time it takes for a packet to travel from $s(r)$ to j along route r, and let T_{jr} be the propagation delay from link j to source $s(r)$, that is the length of time it takes for congestion control feedback to reach $s(r)$ from j along r. (It is assumed that a packet must reach its destination before an acknowledgment containing congestion feedback is returned to its source.) The *round trip time* for route r is then given by $T_r = T_{rj} + T_{jr}$ for all $j \in r$. Finally, we use the notation $a = [\, b \,]_c^+$, defined for $c \geq 0$, to mean $a = b$ if $c > 0$ and $a = \max(0, b)$ if $c = 0$.

2.2 Multipath rate allocation problem

A flow between each source and destination is split between a collection of, n, subflows which use various paths and thus share network resources. A multipath TCP controller can be applied as a pricing and congestion response mechanism in order to dynamically control rate allocations between subflows in a coordinated but still decentralized fashion.

For each route r there is an associated flow rate $x_r(t) \geq 0$, which represents a dynamic fluid approximation to the rate at which the source $s(r)$ sends packets along the route r at time t.

We use the above network description and consider the fluid model of multipath routing extension to TCP given in [11] briefly summarising its primal version here. Further discussion of fluid models for multipath TCP can be found in [8,12] and [2] investigated their use in studying multipath routing in hybrid networks. In this fluid-flow model of joint routing and rate control, the flow rates, $x_r(t)$, vary according to the following differential equations:

$$\dot{x}_r(t) = \frac{x_r(t - T_r)}{T_r} \left[\, \overline{a}(1 - \lambda_r(t)) - b_r T_r y_{s(r)}(t) \lambda_r(t) \, \right]^+_{x_r(t)}, \tag{1}$$

where

$$\lambda_r(t) = 1 - \prod_{j \in r} \left(1 - \mu_j(t - T_{jr})\right), \; y_s(t) = \sum_{\tilde{r} \in s} x_{\tilde{r}}(t - T_{\tilde{r}}) \tag{2}$$

and

$$\mu_j(t) = p_j \left(\sum_{\tilde{r}: j \in \tilde{r}} x_{\tilde{r}}(t - T_{\tilde{r}j}) \right). \tag{3}$$

In the above formulation $\overline{a}$, b_r and functions $\lambda_r(\cdot)$ $(r \in R)$, $\mu_j(\cdot)$, and $p_j(\cdot)$ $(j \in J)$ have the following interpretations. Each link j has a capacity $C_j > 0$ and exhibits congestion by dropping or marking packets via a penalty function given by

$$p_j(z_j) = \left(\frac{z_j}{C_j} \right)^{\beta_j} \tag{4}$$

for some constant $\beta_j > 0$ (known as the *link responsiveness*). Thus, $\mu_j(t)$ is the link j's dropping/marking rate at time t and $\lambda_r(t)$ is the route r's proportion of the acknowledgements that indicate congestion. The quantity $\overline{a}$ is a proportionality factor of the amount by which the sending rate is increased (on receipt of a positive acknowledgement), and b_r is a route-specific proportionality factor of the amount by which the sending rate is decreased (on receipt of a negative acknowledgement through a timeout and so indicating congestion). This controller takes the following form in terms of a control window update algorithm:

- the algorithm responds to each acknowledgement received in a round trip time in which congestion on route r has not been detected with the update

$$\texttt{cwnd}_r \leftarrow \texttt{cwnd}_r + \bar{a};$$

- upon the first detection of congestion in a given round trip time, the congestion window $\texttt{cwnd}_r$ is reduced as follows

$$\texttt{cwnd}_r \leftarrow \texttt{cwnd}_r - b_r T_r y_{s(r)}(t).$$

We will refer to the above controller with the choice of values $\overline{a} = 0.01$ and $b_r = 0.125$ as scalable Multipath TCP (sMPTCP). For the experiments discussed in Section 4 we use $\beta_j = 8$. This choice of the controller's parameters ensures its stability, as this choice fulfils the sufficient condition for local stability of the above (primal) algorithm with the penalty function given by (4) (see [11]), namely that $\overline{a}(1 + \max_{j \in J} \beta_j) < \frac{\pi}{2}$.

Note that equations (1)-(3) with r spanning the whole set of routes R define a set of coupled first-order non-linear differential equations with discrete time delays. Hence, in order to obtain a particular flow dynamics $x_r(t)$, $r \in R$, in a network topology with at least one shared link one needs to solve the entire system of differential equations.

3. CAPACITY REGION AND BOUNDS

In this section we extend the fluid-level modelling to incorporate the notion of connections arriving with workload requirements corresponding to individual file downloads. Thus there is a need to understand the set of workload arrival rates that are compatible with the available network capacity to handle the offered load. Accordingly, in this section we define the capacity region for our network and precisely characterise the boundary of this region as the solution of an optimisation problem. The optimisation problem takes the form of a linear program and this then forms the basis of a technique to determine the range of traffic arrival rates compatible with stable network operation.

3.1 Determination of capacity region

Consider the description of the network model given in Section 2.1 and further suppose that each directed link $j \in J$ has a capacity for flow given by C_j.

We now suppose that connections to the network arrive according to a stationary stochastic processes of rate ν_s for source-destination pair s independently across pairs. Each individual connection for pair s has a randomly chosen file size of data to be transferred, which we suppose has a mean file size of m_s, where file sizes are determined independently. Thus the rate, ρ_s, of work arriving for source-destination pair s is given by

$$\rho_s = \nu_s m_s \,. \tag{5}$$

Further, let $x_r \geq 0$ be the flow on route r and define the following matrices. Let A_{jr} (for $j \in J$ and $r \in R$) be 1 if link j belongs to route r and 0 otherwise. Let H_{sr} (for $s \in S$ and $r \in R$) be 1 if source-destination pair s uses route r and 0 otherwise. Thus,

$$A_{jr} = \begin{cases} 1 & \text{if } j \in r \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad H_{sr} = \begin{cases} 1 & \text{if } r \in s \\ 0 & \text{otherwise} \end{cases} .$$

Write $\rho = (\rho_s; s \in S)$, $C = (C_j; j \in J)$, $m = (m_s; s \in S)$ and $x = (x_r; r \in R)$. We can now define the *capacity region*, Ω, for feasible workload rates as follows (see Fig. 1):

$$\Omega = \left\{ (\rho_s; s \in S) \in \mathbb{R}_+^{|S|} \,:\, \exists\, x_r \geq 0,\, Ax \leq C,\, \rho \leq Hx \right\}$$

We can construct the boundary of Ω as follows. Suppose that $\overline{\rho}$ is any non-zero vector in $\mathbb{R}_+^{|S|}$. Then scalar multiples, $\alpha\overline{\rho}$, of the vector $\overline{\rho}$ all lie in the direction of the ray $\overline{\rho}$ from the origin and we can determine the boundary of Ω by maximizing α subject to $\alpha\overline{\rho}$ remaining within the capacity region. Accordingly, the maximal value, α^*, say, of the objective function from the following linear program determines the point, $\alpha^*\overline{\rho}$, on the boundary of Ω that intersects the ray $\overline{\rho}$ (see Fig. 1):

$$\text{max} \quad \alpha \tag{6}$$

$$\text{subject to} \quad Ax \leq C\,, \alpha\overline{\rho} \leq Hx\,, x \geq 0\,, \alpha \geq 0. \tag{7}$$

For further examples of similar bounding techniques see [6].

The capacity region Ω describes the feasible set of workload rates $(\rho_s; s \in S)$ that can be supported by the set of routes and link capacities determined by the matrix H and the vector C, respectively. Note that the use of linear programming for determining the boundary of the capacity region has further insightful features such as sensitivity analysis and identifying network bottlenecks in terms of the Lagrange multipliers.

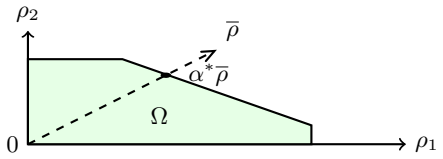

Figure 1: The capacity region Ω describes the set of feasible workload vectors $\rho \in \mathbb{R}_+^{|S|}$.

3.2 Upper bound on arrival rates

We can view our construction of the capacity region Ω as finding the feasible vectors of workload rates ρ constrained in such a way that the workloads are in relative proportions given by the relative proportions of the corresponding components of $\overline{\rho}$. If we take the specific choice $\overline{\rho} = (m_s; s \in S)$ then the point on the boundary, $\alpha^*\overline{\rho}$, will give the maximising scalar α^* the interpretation of an upper bound on the arrival rate common over source-destination pairs. This follows from (5) and since $\alpha^*\overline{\rho} = \alpha^* m$ is the vector of workload rates when average file sizes for different connections are given by the vector m and the arrival rate per connection is α^*.

3.3 Complexity

In our discussion of the capacity region Ω the matrix A captures the link-route incidence relationship, whereas matrix H captures the correspondence between existing routes and available subflows. Hence, both matrices are highly sparse and their dimensions vary with the choice of our parameter n—the number of subflow paths per source destination pair. Specifically, the size of A is $|L| \times |R|$ and the size of H is $|S| \times |R|$. Hence, the size of the linear program (6-7) is $(|L| + |S|) \times (|R| + 1)$, or, $\mathcal{O}(N^2) \times \mathcal{O}(nN^2)$, where N is the size of the set of communicating nodes and n is the number of subflows in use.

3.4 Examples

We shall now consider computing α^* in several illustrative example networks. In our first example we used the network shown in Fig. 2 (referred to as the *single-parented*[1] network) which has $N = 10$ nodes and a total of 30 directed links $j \in J$ with link capacities $C_j = 1$. The nodes are of two types: five source-destination nodes $(n_1, n_2, \ldots, n_5)$ connected with inbound and outbound links to a fully connected core network consisting of five nodes $(n_6, n_7, \ldots, n_{10})$. The set S is the set of distinct node pairs in $\{n_1, n_2, \ldots, n_5\}^2$. The mean files sizes of connections were $m_s = 1$ for all $s \in S$. For this network we take a single subflow, that is $n = 1$, routed along the shortest path (according to hop count). By inspecting the number of subflows present on an individual link we can readily determine that the upper bound on arrival rate is $\alpha^* = 0.25$ since each peripheral node connects to the core through a link of capacity 1 that carries four subflows, one to each of the remaining four peripheral nodes. We also verified this value by computing the bound through the linear programming approach given in Section 3.2.

[1]More generally, *multi-parented* networks have been well-studied for their robustness properties in the context of circuit-switched telecommunication networks, for example see [7].

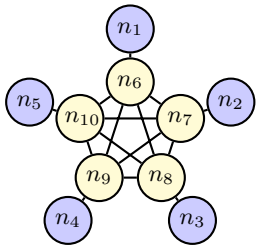

Figure 2: A single-parenting communication network with the complete graph (nodes $n_6, n_7, \ldots, n_{10}$) as its core part.

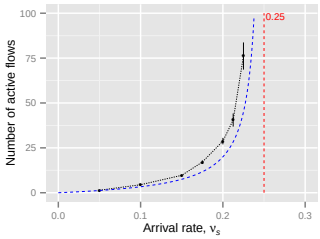

Figure 3: This figure shows the processor sharing approximation to the number of active flows (the blue dashed line) and it's relation to the upper bound α^* on ν_s. Also shown are simulation results for our connection-level fluid model.

Consideration of the number of subflows on a single link also suggests a straightforward approximation for the stochastic process determining the number of flows currently present (termed the *active flows*). Consider a $M/G/1$ processor sharing queue model where jobs in the queue represent active flows for a fixed source node and take the service rate of the queue equal to the common link capacity $C_j = 1$. Such a queue is known to be insensitive to the job size distribution other than to its first moment and for the equilibrium distribution to be geometric with mean $4\nu_s/(1-4\nu_s)$ where ν_s is the arrival rate of connections between a source and destination pair. Here, $4\nu_s$ is the arrival rate of connections to an individual link. Fig. 3 shows the upper bound $\alpha^* = 0.25$ on the arrival rate ν_s obtained by solving the linear program together with the mean number of active flows in the network (as the blue dashed line) given by the expression

$$5 \times \frac{4\nu_s}{1-4\nu_s}, \tag{8}$$

if we also assume independence between the five processor sharing resources.

For fundamental insensitivity results on the processor sharing queue see [15, 16]. In this example we maintain a high degree of symmetry over connections and the processor sharing model seems insightful. More widely, there are natural concerns about the application of (egalitarian) processor sharing to the sharing of resources by TCP flows with differing round trip times and other sources of differentiation: see [1] for a detailed presentation.

Fig. 3 also includes further results obtained by a simulation approach which we will return to in Section 4.

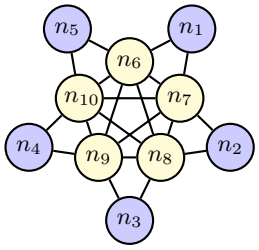

Figure 4: A dual-parenting communication network with the complete graph (nodes $n_6, n_7, \ldots, n_{10}$) as its core part.

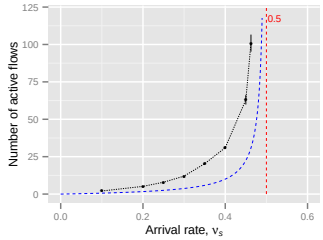

Figure 5: This figure shows the processor sharing approximation to the number of active flows (the blue dashed line) and it's relation to the upper bound α^* on ν_s given by the capacity region for the dual-parented example. Also shown are simulation results for our connection-level fluid model.

Our second example network (referred to as the *dual-parented* network) is shown in Fig. 4. It again contains $N = 10$ nodes with five nodes as a fully connected core but where now each peripheral node connects to two parent nodes as shown. We again take mean file sizes as $m_s = 1$ and link capacities as $C_j = 1$. In contrast with the single-parented network we now allow $n = 4$ subflows for each connection using shortest path routes. Again we can determine the upper bound α^* on the arrival rate ν_s but now we must count subflows. Consider a single source node and note that each peripheral link of capacity 1 now carries two subflows. Thus an individual peripheral link has a mean of $2\nu_s/(1-2\nu_s)$ active subflows. There are 10 such peripheral links and each flow splits into $n = 4$ subflows. Hence, overall, the mean number of active flows is given by the expression

$$\frac{10}{4} \times \frac{2\nu_s}{1-2\nu_s}, \tag{9}$$

which is shown as the blue dashed line in Fig. 5 together with the bound $\alpha^* = 0.5$ verified by our linear programming approach.

In our first two examples we have held the number of subflows fixed at $n = 1$ and $n = 4$, respectively. We now consider several examples where it is natural to vary n and we observe how the capacity region is extended as n increases. Our third example network shown in Fig. 6 contains $N = 6$ nodes and 18 directed links each labelled by their capacity. The choice of links was chosen to give a regular graph where every node has degree 3. We vary n in the range from 1 to 4 and use n shortest paths for the subflows. We computed the

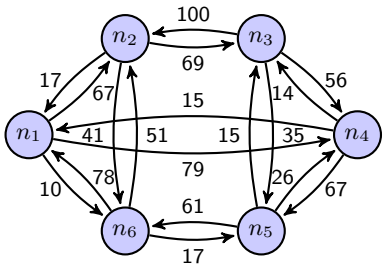

Figure 6: A six node network with 18 directed links labelled by their respective capacities. This network has a regular structure of node degree 3 and there are a total of 30 (ordered) source destination pairs.

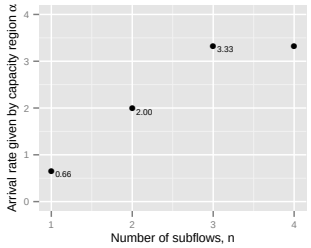

Figure 7: Upper bounds for the arrival rate ν_s with varying parameter n controlling the number of subflows and hence the extent of load balancing across subflow paths.

upper bound α^* using the linear programming technique and our results are shown in Fig. 7. We find that α^* increases with n from 1 to 2 and from 2 to 3 but then remains constant with further increases beyond $n = 4$.

Our fourth example considered a network with $N = 50$ nodes with links chosen to form a regular graph with a node degree of 15. The link capacities were assigned randomly by drawing independent observations from a uniform distribution on the interval 10 to 100. The mean file sizes were again taken to be $m_s = 1$ for all $s \in S$. Although the linear program in this case was substantially larger than that for the first three examples it remained tractable and we were able to compute α^* and obtained the bounds shown in Fig. 8 which increased as far as $n = 6$ before levelling

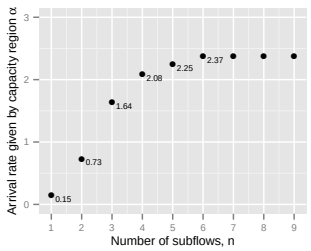

Figure 8: Upper bounds for the arrival rate ν_s in a $N = 50$ node network with regular node degree equal to 15 as the number of subflows, n, varies.

off for n increased further. In this example the set of paths for a particular value of n was found by applying a *hashing* technique designed to balance load across paths at each node when deciding which of the outgoing links to use (see [3] for full details of the `mod k` hashing technique).

The linear program technique is seen to be a useful and flexible tool for investigating the effectiveness of single and multipath protocols. In the next section we will find the bounds insightful regarding our fluid level simulations.

4. EXPERIMENTS

4.1 Simulation of fluid models

In this section we discuss approaches to simulating the stochastic process of connections with random file sizes arriving according to independent Poisson processes. Suppose connections arrive according to a Poisson process of rate ν_s common to all source-destination pairs s. Each connection has an associated random file size drawn independently from a Pareto distribution of mean $m_s = 1$ (specifically with a *location* parameter of 2/3 and a *shape* parameter of 3). The Pareto distribution is frequently chosen as a distribution for filesize downloads. The round-trip times (RTT) for the various routes, T_r, are given by 2×10^{-3} time units per hop (both ways). Thus, for example, a path of length three would have an RTT of 6×10^{-3} time units. Connections for a given source-destination pair use $n \in \{1, 2, 3, 4\}$ multipath subflows.

Although the fluid model with time delayed differential equations provides a detailed fluid approximation of flow dynamics we have found that solving this time delayed version of the fluid model is very computationally intensive. Instead, in a simplified model we shall determine equilibrium sending rates for all connections and suppose that the connections send at these (constant) rates until either a new connection arrives or some active connection reaches the end of its file transfer (whichever occurs first). At such time epochs a single connection is either added or removed from the set of active multipath connections and the equilibrium sending rates are then re-computed before the file transfers for active connections proceed. In this way we have a simplified, piece-wise constant, view of the way in which sending rates vary over time.

Mathematically, the system of delayed differential equations (1-3) is replaced with the simpler system of ordinary differential equations without time delays

$$\frac{dx_r(u)}{du} = f_r\left((x_r(u); r \in R)\right), \tag{10}$$

where the derivative functions f_r are given by modifying (1-3) to reflect the instantaneous rather than the delayed quantities so that

$$f_r\left((x_r(u); r \in R)\right) = \frac{x_r(u)}{T_r}\left[\,\overline{a}(1-\lambda_r(u)) - b_r T_r y_{s(r)}(u)\lambda_r(u)\,\right]^+_{x_r(u)} \tag{11}$$

where

$$\lambda_r(u) = 1 - \prod_{j \in r}\left(1-\mu_j(u)\right),\; y_s(u) = \sum_{\tilde{r} \in s} x_{\tilde{r}}(u) \tag{12}$$

and

$$\mu_j(u) = p_j\left(\sum_{\tilde{r}:j\in\tilde{r}} x_{\tilde{r}}(u)\right), \tag{13}$$

and with the initial conditions $x_r(u)|_{u=0} = x_r(t)$ for all $r \in R$. We have introduced the alternative time variable u in place of t to emphasise that our determination of the long-run stationary solution does not take place over simulated time t but is rather a computation to determine the piecewise constant sending rates $x_r(t) = \lim_{u\to\infty} x_r(u)$. These rates $x_r(t)$ are then updated at time epochs when the number of connections changes and a new equilibrium solution pertains. Numerically we use a fourth order Runge-Kutta method applied to the system (11-13) with a time step increment of $\Delta u = 10^{-2}$ and where the iteration is terminated when $||x_r(u+\Delta u) - x_r(u)||_\infty < 5\times 10^{-4}$ (here $||z||_\infty = \max\{|z_1|, |z_2|, \ldots, |z_n|\}$ denotes the maximum norm for n-dimensional vectors).

For an alternative approach based on multipath dual rather than primal congestion control algorithms see [17].

4.2 Comparison with bounds

Recall our first two example networks from Section 3.4 shown in Fig. 2 and Fig. 4. We simulated the connection-level stochastic processes using the fluid model to allocation sending rates and obtained detailed statistics on the number of active flows present in the network. For the case of the single-parented network (shown in Fig. 2) Fig. 3 shows the estimated mean number of active flows as a function of the arrival rate ν_s as well as the standard errors in our estimates given by the short vertical bars. We can see that the response is similar to our $M/G/1$ processor sharing model with a rapid increase in the number present as ν_s approaches α^* from below.

Similarly, Fig. 5 which shows the estimated mean number of active flows for the dual-parented network together with standard errors. Here the processor sharing model is not quite so accurate but it still captures the essential response as well as the behaviour close to the bound α^*.

Note, however, that the M/G/1 processor sharing model makes a number of assumption which may lead to inaccuracies: (i) the model assumes independence and (ii) the egalitarian sharing of the *entire* available service rate which is unlikely to be achieved by our penalty function approach (see equation 4) in the fluid model.

4.3 Teletraffic models of robustness

Multpath protocols offer several advantages compared to single path protocols but perhaps their most significant benefit is that they introduce a degree of robustness to traffic mismatches and link failures through load balancing. In this section we investigate this type of robustness through study of changes to the capacity region as traffic or link capacity are perturbed.

Consider the network of Fig. 9 with $N = 12$ nodes and 15 directed links. This network was studied in [5] with a static configuration of connections. The capacities of the links are equal to 1 with the exception of the link from n_9 to n_{10} which has a variable capacity of C. There are just three source destination pairs, namely: (n_1, n_4), (n_2, n_5) and (n_3, n_6). Each connection uses $n = 2$ subflows assigned to the two three-hop paths. If $C = 1$ then the pattern of traffic resulting from multipath protocols is symmetric and the capacity region

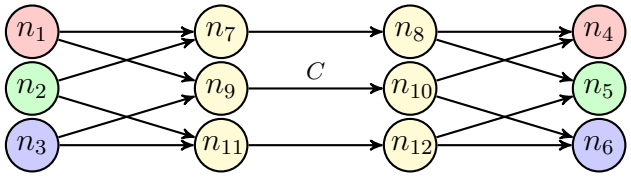

Figure 9: A network with three source destination pairs and 15 directed links.

approach gives a maximum arrival rate of $\alpha^* = 1$. Now consider reducing the capacity by a factor f as f varies from 0 to 1. Fig. 10 shows the effect on the maximal arrival rate for multipath with $n = 2$ and for comparison where at least one of the single paths ($n = 1$) of the three communicating pairs uses link (n_9, n_{10}). We can see that multipath protocols are able to adapt to capacity reductions more easily than single-path protocols. With single-path TCP, as f increases, the single link from n_9 to n_{10} will act as the bottleneck and the achievable arrival rates, ν, will be constrained so that $\nu \le 1 - f$. However, with multipath TCP the cut separating the source nodes from the destination nodes comprising the three links: (n_7, n_8), (n_9, n_{10}) and (n_{11}, n_{12}) acts as the bottleneck and so the achievable arrival rates are instead constrained by $\frac{1}{3}\times(2+(1-f))$, which is greater than $1-f$ for $f > 0$.

Alternatively, capacities may remain fixed but traffic varies to become non-uniform. In order to better understand this effect we consider a model for parametrized traffic mismatch considered in [7]. Consider a parameter δ fixed with $0 \le \delta \le 1$ and then take a uniform traffic matrix of rate ν_0, say, and shuffle it according to the following procedure. Pick two pairs of nodes which act as sources and destinations of traffic chosen equally likely and a random variable uniformly distributed on $[0, \delta\nu_0]$ and swap that amount of traffic between the two pairs of nodes. Now remove those two pairs of nodes from consideration and repeat the procedure with the remaining pairs of nodes swapping randomly chosen traffic between the pairs of nodes but leaving the total traffic fixed. Formally, this traffic reshuffling scheme is outlined in Procedure 1.

Procedure 1 TrafficReshuffle

Input: S, $\nu_0 \ge 0$ and $\delta \in [0, 1]$.
Output: A perturbed traffic matrix $\boldsymbol{\nu}$.

```
Σ ← S
while |Σ| > 1 do
    Choose s1 ∈ Σ and s2 ∈ Σ at random
    Sample U ~ Uniform[0, δν0]
    ν_{s1} ← ν0 − U; ν_{s2} ← ν0 + U; Σ ← Σ \ {s1, s2}
end while
return ν = (ν_s; s ∈ S)
```

Fig. 11 shows the effect this procedure has on the upper bound of the arrival rate given by our linear programming formulation. In the case of the multipath protocol swapping traffic in this way has no effect on the capacity available to a connection but when $n = 1$ and a single subflow the random shuffling of the traffic will reduce the upper bound on the arrival rate. Fig. 11 shows the median rate together with the inter-quartile range of rates shown by the vertical lines.

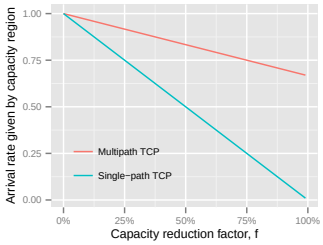

Figure 10: Arrival rate determined by the capacity region with varying middle link capacity.

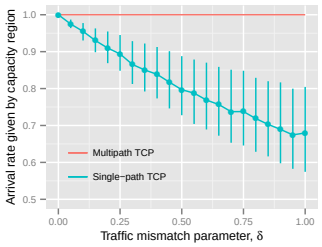

Figure 11: Arrival rate determined by the capacity region with varying traffic mismatch parameter δ.

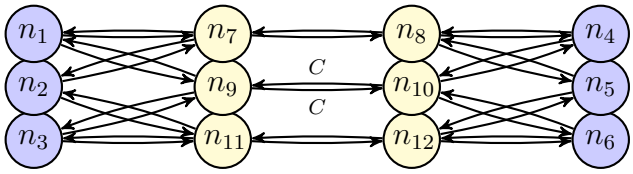

Figure 12: A network with 6 nodes acting as sources and destinations of traffic and with 30 directed links.

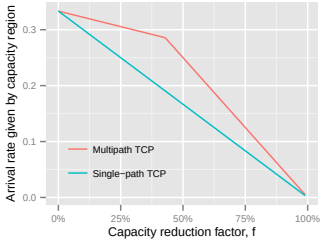

Figure 13: Arrival rate determined by capacity region with varying middle link capacity.

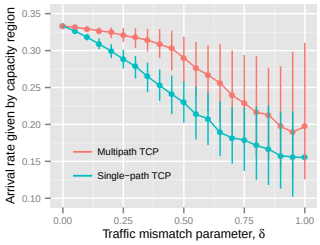

Figure 14: Arrival rate determined by the capacity region with varying traffic mismatch parameter δ.

Fig. 12 shows our second example but now all of the nodes $n_1, n_2, \ldots, n_6$ are sources and destinations. We make the capacities of the links between the yellow nodes equal to 1 except for (n_9, n_{10}) and (n_{10}, n_9) which take the common variable capacity C, $C \in [0, 1]$. All other link capacities are set to a very large number; specifically, the capacity of any link connecting a blue and yellow node equals 10^3. In this example the choice of routes for single paths is as follows:

- link (n_7, n_8) carries traffic for $n_1 \leftrightarrow n_4$, $n_1 \leftrightarrow n_5$, $n_2 \leftrightarrow n_4$;
- link (n_9, n_{10}) carries traffic for $n_1 \leftrightarrow n_6$, $n_3 \leftrightarrow n_4$, $n_3 \leftrightarrow n_6$;
- link (n_{11}, n_{12}) carries traffic for $n_2 \leftrightarrow n_5$, $n_2 \leftrightarrow n_6$, $n_3 \leftrightarrow n_5$.

When $n = 2$, the communicating (blue) nodes are, whenever possible, connected via two different paths, e.g. n_1 connects to n_4 via both (n_7, n_8) and (n_9, n_{10}); however, both multipath subflows connecting n_1 to n_6 use the path $n_1 \to n_9 \to n_{10} \to n_6$.

Applying the above traffic reshuffle scheme detailed in Procedure 1 we obtain Fig. 13 and Fig. 14. In this case, the multipath protocol with $n = 2$ is also affected by the traffic mismatch parameter δ but, as expected, shows a higher degree of robustness compared to single-path TCP. Note also that in this example link (n_9, n_{10}) represents a bottleneck for both single-path and multipath TCP (when connecting between nodes n_1 and n_6 as well as n_3 and n_4), and so one would indeed expect both lines from Fig. 13 to meet when the middle link's capacity reduction factor equals 100%.

5. CONCLUSIONS

In this paper we have considered multipath protocols and constructed a capacity region for achievable workload rates determined by a linear programming approach. The bounds are approached by connection-level simulations of a fluid model for the behaviour of the multipath rate controller. We have further investigated the robustness properties of multipath TCP according to the degree of load balancing introduced by the use of multiple subflow paths. Finally, we have quantified the robustness in situations arising from random traffic mismatches or from link failures.

Acknowledgements

Research was sponsored by US Army Research laboratory and the UK Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and

conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the U.S. Government, the UK Ministry of Defense, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

The authors gratefully acknowledge useful discussions with Frank Kelly.

6. REFERENCES

[1] S. Aalto, U. Ayesta, S. Borst, V. Misra, and R. Núñez Queija. Beyond processor sharing. *SIGMETRICS Perform. Eval. Rev.*, 34(4):36–43, Mar. 2007.

[2] A. Bejan, R. Gibbens, Y. sup Lim, and D. Towsley. A performance analysis study of multipath routing in a hybrid network with mobile users. In *Teletraffic Congress (ITC), 2013 25th International*, pages 1–9, Sept 2013.

[3] Z. Cao, Z. Wang, and E. Zegura. Performance of hashing-based schemes for internet load balancing. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 332–341 vol.1, 2000.

[4] Y.-C. Chen, Y.-s. Lim, R. J. Gibbens, E. M. Nahum, R. Khalili, and D. Towsley. A measurement-based study of multipath TCP performance over wireless networks. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, pages 455–468, New York, NY, USA, 2013. ACM.

[5] R. J. Gibbens. Modelling multi-path problems. In *Proceedings of 42nd Annual Conference CISS 2008, Information Sciences and Systems*, pages 42–45, 2008. Princeton, NJ.

[6] R. J. Gibbens and F. P. Kelly. Network programming methods for loss networks. *IEEE Journal on Selected Areas in Communications, invited paper for special issue on Advances in the Fundamentals of Networking*, 13(7):1189–1198, 1995.

[7] R. J. Gibbens, F. P. Kelly, and S. R. E. Turner. Dynamic routing in multiparented networks. *IEEE/ACM Transactions on Networking*, 1(2):261–270, Apr. 1993.

[8] H. Han, S. Shakkottai, C. Hollot, R. Srikant, and D. Towsley. Multi-path TCP: A joint congestion control and routing scheme to exploit path diversity in the internet. *Networking, IEEE/ACM Transactions on*, 14(6):1260–1271, Dec 2006.

[9] B. Jiang, Y. Cai, and D. Towsley. On the resource utilization and traffic distribution of multipath transmission control. *Perform. Eval.*, 68(11):1175–1192, Nov. 2011.

[10] F. Kelly. Fairness and stability of end-to-end congestion control. *European Journal of Control*, 9:149–165, 2003.

[11] F. Kelly and T. Voice. Stability of end-to-end algorithms for joint routing and rate control. *SIGCOMM Comput. Commun. Rev.*, 35(2):5–12, Apr. 2005.

[12] T. Kelly. Scalable TCP: Improving performance in highspeed wide area networks. *SIGCOMM Comput. Commun. Rev.*, 33(2):83–91, Apr. 2003.

[13] P. Key, L. Massoulié, and D. Towsley. Path selection and multipath congestion control. *Commun. ACM*, 54(1):109–116, Jan. 2011.

[14] R. Khalili, N. Gast, M. Popovic, U. Upadhyay, and J.-Y. Le Boudec. Mptcp is not pareto-optimal: Performance issues and a possible solution. In *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '12, pages 1–12, New York, NY, USA, 2012. ACM.

[15] L. Kleinrock. Time-shared systems: A theoretical treatment. *Journal of the ACM*, 14(2):242–261, April 1967.

[16] L. Kleinrock. *Queueing Systems*, volume II: Computer Applications. Wiley Interscience, 1976.

[17] B. McCormick, F. Kelly, P. Plante, P. Gunning, and P. Ashwood-Smith. Real-time alpha-fairness based traffic engineering. In *Workshop on Hot Topics in Software Defined Networking (HotSDN)*. ACM SIGCOMM, Aug. 2014.

[18] C. Paasch. *Improving Multipath TCP*. PhD thesis, Universite Catholique de Louvain, Louvain-la-Neuve, Belgium, 2014.

[19] C. Paasch and O. Bonaventure. Multipath TCP. *Communications of the ACM*, 57(4):51–57, Apr 2014.

[20] F. Paganini, Z. Wang, J. C. Doyle, and S. H. Low. Congestion control for high performance, stability, and fairness in general networks. *IEEE/ACM Trans. Netw.*, 13(1):43–56, Feb. 2005.

[21] A. Papachristodoulou, L. Li, and J. C. Doyle. Methodological frameworks for large-scale network analysis and design. *SIGCOMM Comput. Commun. Rev.*, 34(3):7–20, July 2004.

[22] C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley. Improving datacenter performance and robustness with multipath TCP. In *Proceedings of the ACM SIGCOMM 2011 Conference*, SIGCOMM '11, pages 266–277, New York, NY, USA, 2011. ACM.

[23] S. Shakkottai and R. Srikant. Network optimization and control. *Found. Trends Netw.*, 2(3):271–379, Jan. 2007.

[24] R. Srikant. *The Mathematics of Internet Congestion Control (Systems and Control: Foundations and Applications)*. SpringerVerlag, 2004.

[25] D. Wischik, M. Handley, and C. Raiciu. Control of multipath TCP and optimization of multipath routing in the internet. In *Proceedings of the 3rd Euro-NF Conference on Network Control and Optimization*, NET-COOP '09, pages 204–218, Berlin, Heidelberg, 2009. Springer-Verlag.

[26] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley. Design, implementation and evaluation of congestion control for multipath TCP. In *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation*, NSDI'11, pages 99–112, Berkeley, CA, USA, 2011. USENIX Association.