Evaluating the performance of multi-path routing and congestion control in presence of network resource management

Luca Muscariello[1] and Diego Perino[1,2]
[1]Orange Labs Paris, France Télécom R&D - [2] INRIA Project Team "GANG", France
Email: firstname.secondname@orange-ftgroup.com

Abstract—**Network traffic is increasing in size and is becoming more and more dynamic leading to unpredictable and highly variable traffic patters. Multi-path routing is a valuable on-line technique to deal with such trend, mostly for intra-domain TE, multi-homing, wireless mesh networks, metropolitan access networks and has been shown efficient for a large spectrum of future traffic scenarios. In this paper we present MIRTO, a distributed multi-path routing protocol that jointly uses best path selection and flow control for optimality and stability. MIRTO is feedback based and is designed around some key observations inferred from optimization. We analyze MIRTO's performance and we compare it with TEXCP and TRUMP, two other recently proposed multi-path routing algorithms. On a US-like backbone network, with and without in-network fair queuing schedulers, our algorithm proves to work as better as the other two while relying on simpler feedbacks and consuming less network resources. Modeling and analysis of such algorithms is performed through fluid models, based on ordinary differential equations (ODEs).**

I. Introduction

The evolution of network applications, services and devices is modifying the way the Internet is perceived and is having a strong impact on network traffic. As an example, the widespread usage of application-layer overlays (e.g. [1]–[4]) and the increasing popularity of User Generated Content services (e.g. [5]) are increasing the amount of traffic generated by customers which results in more dynamism. This process might be exacerbated in the next years because of the wide deployment of high rate access lines. This will tempt every customer to run bandwidth consuming applications as high rate video or massive gaming (Internet video game tournaments).

For ISPs, service overlays and virtual networks are rapidly gaining popularity thanks to their ease to build and deploy new services. Up to now, they have ignored dynamic optimal network resources allocation whilst focusing more on service separation and configuration. The potential network instability induced by the interaction among multiple overlays and virtual networks, with the consequent unpredictable resources allocation, might diminish the initial enthusiasm on these technologies.

Traditional Traffic Engineering (TE) is based on off-line optimization and occurs at time-scale of hours. Considering the aforementioned traffic trends, off-line TE will lead to an inefficient usage of network resources resulting in expensive link upgrades. Moreover, off-line TE suffers of traditional issues like line-card failures, large flow reroutes due to inconsistent routes' ranking coming from different routing protocols just to cite the most known.

Robust routing exploiting multiple paths is a powerful approach for overload control to deal with the aforementioned traffic patterns, mostly for intra-domain TE, inter-AS path selection under the same ISP, routing in wireless networks and metropolitan access networks. In [26] multi-path routing is shown to be efficient within highly loaded access network for several possible traffic patterns generated by users with different access lines. At light loads traffic needs not to be load balanced among multiple routes.

Multi-path routing can be performed on individual end-to-end flows or on aggregate flows between network nodes at the edge of an AS or between wireless backhauling nodes. For aggregates, time-scales are sufficiently larger than the round trip times (RTT) while for flows time-scales are comparable to RTTs. Fairness plays also a different role in these two scenarios. There is no notion of fairness for aggregates as the objective is to switch the maximum throughput subject to network costs, while for flows fairness is an issue. Scalability concerns arise for flow multi-path routing due to the fact that per flow agents should run at a gateway with multi-path capabilities. This may be solved in case agents are installed at clients, raising issues on cheating sources non respecting a common fairness criterion. Fairness might be imposed by in-network link scheduling like fair queuing or other approximate fair dropping mechanisms (as [28]). Fair queuing (FQ) has many known desirable effects on resource allocation ranging from fast rate convergence to insensitivity to a common fairness criterion in transport protocols. This allows new applications to be easily deployed with no need to handle fairness issues, see [14], [15], with scalable mechanisms ([22], [23]).

In this paper we consider a general routing framework that can be adapted according to the application, i.e. whether the protocol would route aggregates or flows. We propose a new network routing protocol MIRTO, designed around some key observations inferred from the optimization analysis presented in [26]. Our routing scheme, that exploits path diversity, is distributed, adaptive and does not assume infinitely backlogged demands. We introduce an analytical model in order to compare different routing schemes using fluid ordinary differential equations (ODEs). The sending rate of MIRTO and of other two recent algorithms, TEXCP [17] and TRUMP [13], are modeled and compared on the Abilene network topology with FIFO and FQ scheduling. To the best of our knowledge, this is the first time a comparison of the aforementioned routing algorithms is performed via fluid models. We show our algorithm performs as better as the others while relying on simpler feedbacks and consuming less network resources. MIRTO at light loads stops to use multiple routes and converge to shortest path routing. The modeling framework is also a contribution in itself.

The rest of the paper is organized as follows. Section II includes the problem formulation and related work. In section III we present the network protocol MIRTO and in sec. IV the model of sources and network queues for FIFO and FQ scheduling. Section V presents a fluid model for TEXCP and TRUMP and a comparison of the three algorithms. Section VI concludes the paper.

II. Problem formulation and related work

Optimized multi-path routing is an old problem that spurred significant research around it. The first formulation dates back to the work of Gallager on minimum cost routing of datagram's flow, along multiple routes [9]. The mathematical formulation is that of a multi-commodity flow problem, with convex objectives and linear constraints. If the network has enough capacity, traffic demands are optimally routed, otherwise the problem is not feasible. This work evolved towards the joint routing and flow control framework, formalized by Golestaani and Gallager in late seventies (see [10], [11]), that handles network rate limitations, using both path costs and flow rate adaptation. Kelly tackles the problem under a different perspective ([18], [19]), by introducing the concept of user's utility. The focus is on rate control and fairness through the study of stability of differential equations in presence of network delays, see also [30]. A thorough description of recent approaches necessitates a formal definition of the problem.

A. Notation

The network topology is modeled by a connected graph $G = (\mathcal{N}, \mathcal{L})$ given as a set of nodes and links. Be $[a_{ij}]$ the adjacency matrix, $a_{ij} = 1$ if there exists a directional link between i and j and $a_{ij} = 0$ otherwise. The network carries traffic generated by a set of demands $\mathcal{D}$, each characterized by triple (s^d, e^d, p^d), with $s \in \mathcal{S}$, $e \in \mathcal{E}$, source and destination nodes, $\mathcal{S}, \mathcal{E} \subseteq \mathcal{N}$, and $p \in \mathbb{R}^+$ the exogenous peak rate. This latter is the rate the flow would attain if the network had infinite resources. In the model a network flow d, $d \in \mathcal{D}$, gets a share x_{ij}^d of the capacity C_{ij} at each link with $0 \leq x_{ij}^d \leq \min(C_{ij}, p^d)$.
The network flow can be split among different paths that are made available by a network protocol at an ingress node. We make no modeling assumptions on whether paths are disjoint, however the ability to create more path diversity helps design highly robust network routing protocols.

B. User utility and network cost

Utility of a flow d, U^d (assumed strictly convex) and network cost V (assumed strictly concave) are two conflicting objectives in a mathematical formulation. U^d is a function of the total rate y^d, attained by a flow, split among different paths, and is supposed to be an alpha fair function [11], [25]. The cost $V()$ is a function of the link load.

The common mathematical *arc-node* formulation is as follows :

$$\begin{aligned}
&\text{maximize} \sum_{d \in \mathcal{D}} U^d(y^d) - \sum_{i,j \in \mathcal{N}} V(\sum_{d \in \mathcal{D}} x_{ij}^d / C_{ij}) \\
&\text{subject to} \\
&\sum_{k \in \mathcal{N}} a_{ik} x_{ki}^d - \sum_{j \in \mathcal{N}} a_{ji} x_{ij}^d = \phi_i(y^d) \quad \forall i \in \mathcal{N}, d \in \mathcal{D} \\
&\sum_{d \in \mathcal{D}} x_{ij}^d \leq C_{ij} \quad \forall i,j \in \mathcal{L}, d \in \mathcal{D} \\
&y^d \leq p^d \quad \forall d \in \mathcal{D}.
\end{aligned} \tag{1}$$

where $\phi_i(y^d)$ is the flow in each node i that is zero in every forwarding node, equal to y^d or $-y^d$ in a source or destination node respectively. In the *path-demand* formulation, the flow rate y^d of a given demand d can be rewritten as the sum of the rates over available paths $\mathcal{P}^d$, i.e. $y^d = \sum_{i \in \mathcal{P}^d} y_i^d$. In such a formulation a user coordinates sending rates over paths jointly, aiming at maximizing its own utility.
Authors in [13], [17] have used the cost function to take into account TE objectives usually representing link load minimization, i.e. expenditures for line-card upgrades. [12], [13], [17], [20] also propose distributed algorithms aiming to solve the global optimization problem usually using decomposition techniques.

In our previous work [26] we interpret network cost V as to select minimum cost routes (path ranking function) and we consider max-min as fairness criterion. This allows us to solve large problems in realistic networks and so to quantify the gain in deploying multi-path routing in future network architectures. We observe that, under these constraints, flows first fill their best paths and exploit their secondary paths only if some bandwidth demand remain unsatisfied and alternative paths have some remaining resources. This quite simple criterion does not show up in the case all demands have no rate limitations, which is a common assumption in the literature for simplicity.

Starting from these observations, in the next section we introduce MIRTO, which is a distributed algorithms built on the water filling procedure of max-min fairness, as basic criterion.

III. MIRTO, the network protocol

Multi-path iterative routing traffic optimizer, or just MIRTO, is designed to route traffic from an ingress node of a network toward an egress node, traversing a number of forwarding nodes. A MIRTO agent runs at the ingress node (access gateway) and is responsible to split the incoming traffic over multiple paths toward the destination. The sending rate over each path is updated every T according to Algorithm 1. A MIRTO agent can be associated to a flow or to a traffic aggregate according to the considered application context (TE, users' flow routing, multi-homing etc.).

Every node in the network implements signalization functionalities to maintain updated a series of structural network informations, such as paths state and ranking, and is responsible for the maintenance of routing tables and forwarding paths.

Every egress node has to feed back to the sources whether used paths are congested or not. Note that in presence of Fair Queuing schedulers (only in case of per flow routing) the congestion notification is per flow and selectively sent to those flows that are causing congestion. A path is considered congested by a MIRTO agent when at least one congestion notification has been received during last T.

A. Path selection and congestion control

Following the water filling procedure of max-min fairness (see [26]) MIRTO starts to fill the best ranked paths first. Every T a rate increase is performed over the best path until the demand reaches its backlog or the path maximum share is attained. In the first case, it means that there is no need to look for bandwidth elsewhere as satisfaction has attained the maximum. In the second case, MIRTO starts to seek resources from the second best ranked path. This procedure iterates over the other paths and stops whether the demand attains its peak rate (i.e. before using all paths), or whether the path maximum share is attained over all paths. Every T a rate decrease is performed over all congested paths.

B. Probing network resources

The above described procedure implies that all MIRTO transmitters implement a common per path rate fairness criterion. MIRTO uses a window flow control protocol additive increase multiplicative decrease (AIMD) as TCP. The particular chosen fairness criterion is out of our scope, while it is fundamental to pick the same for all MIRTO agents. The minimum requirement is that egress nodes feed back to the sender paths' state. Enhanced congestion notifications are possible as ECN or RE-ECN (see [7]).

C. Preventing non optimal allocations

AIMD in presence of drop tail buffer management is known to have no stationary solution (the sawtooth TCP regime). Therefore, the average rate is the mean over a periodic cycle. Hence, it might happen that the attained rate, along a given path, oscillates around the mean. As a matter of fact, it is unwise relying on oscillations, as they depend on many factors: buffer sizes, RTTs, synchronization etc. MIRTO estimates the attained rate using a moving average that filters out high frequencies to detect if available bandwidth is truly varying. The cut-off scale of the filter should be chosen much larger than the round trip time over the path. We say that a path i is in *steady state* with respect to a source d if the mean sending rate of flow d along path i varies with respect to its absolute value for less than a few percent.

Once all paths get to a *steady state* an allocation is attained which might be globally non optimal. In order to guarantee MIRTO to find optimal or nearly optimal allocations we must avoid that the probing procedure is trapped in such a condition. Therefore MIRTO exits this state reducing the rate over all paths. To summarize, the protocol pseudo-code is reported in algorithm 1.

Algorithm 1 MIRTO: Every T

```
if $\exists y_i^d$ : NotCongested( $y_i^d$ ) and NotSteady($y_i^d$) then
    for $j \in \mathcal{P}_d$ do
        if isCongested ($y_j^d$) then
            $y_j^d = y_j^d - \beta y_j^d$
        else if isSteady( $y_j^d$ ) and isBestSteadyPath( $y_j^d$ ) then
            $y_j^d = y_j^d + \alpha$
        else if isNotSteady( $y_j^d$ ) and isBestNonSteadyPath( $y_j^d$ ) then
            $y_j^d = y_j^d + \alpha$
        end if
    end for
else
    for $j \in \mathcal{P}_d$ do
        $y_j^d = y_j^d - \gamma \sum_j y_j^d$
    end for
end if
```

D. Limited backlog at sources

Assuming demands have infinite backlog, rate increase is always possible. If the demand is rate limited, it frequently happens that the source has no sufficient backlog to send. The effective rate increase is given by the minimum between the potential rate and the actual backlog. This regularly happens in the current Internet in case an application is not able to feed the TCP sending buffer at the network rate. Other important scenarios exhibit such behaviors, like multi-hop TCP and split TCP [8]. In all cited cases the rate increase will be sub-linear for AIMD up to the peak rate of the source. In [8] authors study analytically this phenomenon in case the slow application is an upstream slower TCP connection that might not feed the second one at the requested rate.

Rate limited transmitters would probe the network state at a slower rate. This also means that the fairness criterion of rate limited flows is slightly changed with respect to the original utility function.

IV. Fluid modeling of MIRTO and queues

We model each MIRTO source using a fluid representation of the sending rate $y_i^d(t)$ along each path i for a given flow d. We suppose sources use an AIMD flow controller and we take delays into account. Therefore, rates' evolution is described through a system of deterministic ODEs along the line of classical fluid models of TCP [24], [29]. Let us write $R_i = T \vee RTT_i(t)$ where $1/T$ is the probe rate. For all y_i^d,

$i \in \mathcal{P}_d$, $d \in \mathcal{D}$ (being $\mathcal{P}_d$ the set of paths of flow d and $\mathcal{D}$ the set of flows) we have:

$$\frac{dy_i^d(t)}{dt} = \frac{\alpha\psi_i^d(t)}{R_i^2} - \beta y_i^d(t)\phi_i^d(t-R_i) - \gamma \sum_{j\in\mathcal{P}_d} y_j^d(t)\zeta^d(t-R_i) \tag{2}$$

Let us illustrate each term in detail.

1) Increase term: the first term, at the right member of (2) accounts for the additive increase of $y_i^d(t)$ over time with slope α/R_i^2 where α is the increase parameter (=1 in TCP Reno). The increase takes place when the path is selected, according to the decision function $\psi_i(t)$.

$$\psi_i^d(t) = \begin{cases} \prod_{j\in\mathcal{P}_d\setminus i} \ {}_{\{S_i^d<S_j^d\}} & \text{if } i \in \widetilde{\mathcal{P}}_d \\ \prod_{j\in\mathcal{P}_d\setminus\widetilde{\mathcal{P}}_d, j\neq i} \ {}_{\{S_i^d<S_j^d\}} & \text{if } i \in \mathcal{P}_d\setminus\widetilde{\mathcal{P}}_d. \end{cases} \tag{3}$$

where $\widetilde{\mathcal{P}}_d$ defines the set of all paths in "steady state" and $S_i^d(t)$ is a path cost measure defined as the sum of the inverse of the link capacities:

$$S_i^d(t) = \begin{cases} \sum_{k\in\mathcal{L}_i^d} \frac{1}{C_k} & \forall\ k \in \mathcal{L}_i^d \quad Q_k(t-R_i) < B_k \\ \infty & \exists\ k \in \mathcal{L}_i^d \quad Q_k(t-R_i) = B_k. \end{cases} \tag{4}$$

$Q_k(t)$ denotes the size of queue k at time t. As one can remark from (3), the decision function acts differently on paths that are in transitory or in steady state (the definitions of these terms will be precisely defined later). The path selection described in 3, 4 obeys to these rules:

- As long as path i is in steady state, y_i^d grows if and only if i is the minimum cost path among all.
- When path i is transient, y_i^d grows if i is the best path among all transient paths.

Path i is assumed to be congested if at least one link k, $k \in \mathcal{L}_i^d$ is in saturation, i.e. $Q_k(t) = B_k$ being B_k the buffer size and $\mathcal{L}_i^d$ the link set of demand d over path i. The queue models will be presented later. The cost associated to a non congested path i, S_i^d is given by the sum of the inverse of capacities of links in $\mathcal{L}_i^d$ (4), i.e. the sum of the link costs.

2) Decrease terms: the second and the third term at the right member of (2) account for the rate decrease. A congestion notification on a link within path i causes a a rate reduction of $\beta y_i^d(t)$ ($\beta = 1/2$ in TCP Reno). $\phi_i^d(t)$ indicates the occurrence of congestion within path i for flow d, as

$$\phi_i^d(t) = 1 - \prod_{k\in\mathcal{L}_i^d} \ {}_{\{Q_k(t)<B_k\}}$$

regardless of the queue model that we introduce in the next paragraph. In addition to the multiplicative decrease of $y_i^d(t)$ our algorithm introduces a coordinated reduction of the rate, through the term $\gamma\sum_{j\in\mathcal{P}^d} y_j^d(t)$, proportional to the total rate of flow d, when all paths are either congested or in steady state, as expressed by $\zeta^d(t)$,

$$\zeta^d(t) = \prod_{j\in\mathcal{P}_d\setminus\widetilde{\mathcal{P}}_d} \ {}_{\{S_j^d(t)=\infty\}}$$

$\zeta^d(t)$ is conventionally set to one if $\mathcal{P}_d\setminus\widetilde{\mathcal{P}}_d \neq \emptyset$.
The decrease parameter γ quantifies the level of coordination between all paths of a given flow d as it intervenes on all paths jointly. Note that β and γ must be chosen as to avoid the rate to become instantaneously negative. We get rid of this condition in numerical evaluations by limiting the decrease term to $y_i^d(t) \wedge \left(\beta y_i^d(t) + \gamma\sum_{j\in\mathcal{P}_d} y_j^d(t)\right)$. Let us get back to the definition of "steady state". It is worth observing that the increase/decrease dynamics of the rate lead to an oscillatory stationary regime well known for TCP Reno (and other TCP flavours) under drop tail. Thus, we define a flow to be stationary as long as the variations of its mean value remain bounded by a constant ε, i.e.

$$\left|\frac{\tilde{y}_i^d(t) - \tilde{y}_i^d(t-T^d)}{\tilde{y}_i^d(t-T^d)}\right| < \varepsilon,$$

$\tilde{y}_i^d(t)$ denotes the exponential moving average up to time t with smoothing parameter T_i^d, taken proportional to $R_i(t)$, $d\tilde{y}_i^d(t)/dt = -[\tilde{y}_i^d(t) - y_i^d(t)]/T^d$. For bottlenecked sources, equation (2) is slightly modified with an additional term at the right member: a decrease term equal to $\alpha/R_i(t)^2$ over the most expensive path with respect to the previously described definition of cost. This means that, as a path is congested, the same amount of rate is moved from a bad path to a better one. The presence of peak rates avoid a flow d probing a path indefinitely, as this would not result in any benefit.

A. Queue models

The queue models that we consider are FIFO with drop tail and FQ with drop from the longest queue first (DLQF). FIFO is used as a neutral scheduler that does not realize any particular form of fairness because it delegates this issue to end-to-end protocols. FQ, on the contrary imposes at every link max-min fairness among all flows crossing it ([14], [15]).

1) FIFO: The time evolution of $Q_k(t)$ follows:

$$\frac{dQ_k(t)}{dt} = A_k(t) - D_k(t) - L_k(t) \tag{5}$$

where each term is defined according to

$$\begin{aligned} A_k(t) &= \ \sum_{d\in\mathcal{D}} r_k^d y_k^d(t) \\ D_k(t) &= \ C_k \ {}_{\{Q_k(t)>0\}} \\ L_k(t) &= \ (A_k(t) - C_k)^+ \ {}_{\{Q_k(t)=B_k\}} \end{aligned}$$

The arrival rate $A(t)$ is the superposition of all flow rates routed through the queue $Q(t)$. The departure rate $D(t)$ is given by the link capacities when the queue is non empty and zero otherwise. The loss rate $L(t)$ is given by the excess rate $A(t) - C$ in congestion: $A(t) > C, Q(t) > B$, with B the storage capacity. r_k^d is one if flow d is routed through link k, zero otherwise.

2) FQ: Time evolution of Q_k^d, the per flow occupation, is driven by the following set of equations

$$\frac{dQ_k^d(t)}{dt} = A_k^d(t) - D_k^d(t) - L_k^d(t), \quad \forall d \in \mathcal{D} \tag{6}$$

where each term is defined according to

$$\begin{aligned}A_k^d(t) =& r_k^d y_k^d(t)\\ D_k^d(t) =& C_k \frac{\{Q_k^d(t)>0\}}{\sum_{d':r_k^{d'}>0} \quad \{Q_k^{d'}(t)>0\}}\\ L_k^d(t) =& (A_k(t) - C_k)^+ \quad \{d=\arg\max_{d'} Q_k^{d'}(t), \sum d' Q_k^{d'}(t)=B_k\}\end{aligned}$$

$A^d(t)$ is the flow rate of flow d, $D_k^d(t)$ the rate scheduled to flow d and $L_k^d(t)$ the loss rate experienced by flow d, according to the longest queue drop policy.

V. Performance evaluation

In this section we study the performance of MIRTO by means of numerical evaluations of the equations derived in section IV. We compare it to other two recently proposed algorithms TEXCP and TRUMP that are briefly described and modeled in the following. We keep, for each algorithm, the same notation of the corresponding original paper in order to help the reader comparing these equations with the protocol definition. The following notation is valid only within the scope of this section and must not be compared with that of the rest of the paper.

A. TEXCP

TEXCP is a distributed algorithm that balances load over multiple paths trying to minimize the maximum link load over the network.This algorithm is described in [17], and performance are evaluated through simulations and some modeling. TEXCP takes into account links' utilization that is measured in every node and fed back to transmitters through periodic probes of period T_p. Load balancing is adapted every $T_d \geq T_p$ with the following rule, for a flow s on path p:

$$\Delta x_{sp} = \begin{cases} \frac{r_{sp}}{\sum_i r_{si}} \left(\frac{\sum_i x_{si} u_{si}}{\sum_i x_{si}} - u_{sp} \right) + \epsilon & u_{sp} = u_{\min} \\ \frac{r_{sp}}{\sum_i r_{si}} \left(\frac{\sum_i x_{si} u_{si}}{\sum_i x_{si}} - u_{sp} \right) & u_{sp} \neq u_{\min} \end{cases}$$

where r_{sp} is the sending rate of flow s along path p, u_{sp} is the most recent notified utilization along the path p (every T_p), $u_{\min} = \min_p u_{sp}$ and ϵ is a small constant. The resulting split ratio may need to be re-normalized so that $\sum_p x_{sp} = 1$. r_{sp} is the result of flow sharing at the bottleneck along the path using an AIMD rate controller that receives every T_p congestion feedbacks from the nodes. Rate adaptation is then obtained as the weighted difference (by the parameters α and β) between positive and negative feedbacks. An additional parameter, γ may be used to weight rate allocations inversely proportional to the path length or delay (TEXCPSP). See [17] for more details.

We model TEXCP (or TEXPSP) as follows. Consider a given link l, and P^l the set of all flows crossing the l of number $N_l = |P^l|$, be $Q_l(t)$ the queue length, and $u_l = \frac{\sum_{k \in \mathcal{P}^l} r_k}{C_l}$ the link utilization. We write the load balancing equation as:

$$\frac{dx_{sp}}{dt} =$$
$$\begin{cases} \frac{r_{sp}}{\sum_i r_{si}} \left(\frac{\sum_i r_{si}(t) u_i(t-T_p)}{\sum_i r_{si}(t)} - u_i(t-T_p) \right) + \epsilon & u_i = u_{\min} \\ \frac{r_{sp}}{\sum_i r_{si}} \left(\frac{\sum_i r_{si}(t) u_i(t-T_p)}{\sum_i r_{si}(t)} - u_i(t-T_p) \right) & u_i \neq u_{\min} \end{cases}$$

Let us write g_{sp} rate adaptation of flow s over path p, through δ^+ and δ^- the positive and negative feedbacks respectively.

$$\frac{dg_{sp}}{dt} = \delta^+ - \delta^- g_{sp}(t - T_p)$$

where

$$(\delta^+, \delta^-) = \begin{cases} (\frac{\phi_l}{N_l}, 0) & \phi_l \geq 0 \\ (0, \frac{\phi_l}{\sum_{k \in \mathcal{P}^l} r_k}) & \phi_l < 0 \end{cases}$$

being $\phi_l = \alpha \left(C_l - \sum_{k \in \mathcal{P}^l} r_k \right) - \beta Q_l(t)$.
The sending rate over path p of flow s is then $r_{sp} = \min(x_{sp} P_s, g_{sp})$ being P_s the peak rate of flow s.

In order to make TEXCP give priority to the shortest paths (TEXCPSP) an additional variable is required: $v_l^d = (2/R_l^d)^\gamma$ so that $\delta^+ = (\phi v_l^d) / \sum_d v_l^d$ where R_l^d is the RTT of flow d traversing link l.

B. TRUMP

TRUMP is a network protocol that is obtained through decomposition of the optimization problem of routing and congestion control with network costs (1). The objective is to maximize $\sum_s U_s - w \sum_l C_l$, the weighted difference among the aggregated utility among all sources s and aggregated cost among all network links l. Details on the decomposition can be found in [13]. The resulting protocol requires nodes to evaluate the following link congestion measures:

$$\begin{aligned} p(t+T) &= [p_l(t) - \beta(C_l - N_T/T)]^+ \\ q_l(t+T) &= w/C_l \exp\left(\frac{N_T/T}{C_l}\right) \\ s_l(t+T) &= q_l(t+T) + p(t+T) \end{aligned}$$

where N_T is the amount of bits that crossed the link during the time interval T. s_l is then fed back to sources. The rate variation at the sender d along path p is calculated through the following formula:

$$\Delta y_p^d = \gamma \left(1/\sum_{l \in P} s_l - \sum_p y_p^d \right)$$

If the sender has limited backlog the formula is undetermined. Therefore we can only use TRUMP whether demands have no rate limitations.

The dynamic of TRUMP is given by the following ODEs at link l,

$$\begin{cases} p_l(t) = p_l(0) + \max\left(0, -\int_0^t \beta(C_l - \sum_d y_l^d(t)) dt\right), \\ d_l(t) = \frac{w}{C_l} e^{\frac{\sum_d y_l^d(t)}{C_l}}, \quad s_l(t) = p_l(t) + d_l(t) \end{cases}$$

and for a source d with RTT R_i along path i,

$$\frac{dy_i^d}{dt} = -\gamma \sum_i y_i^d(t) - \frac{\gamma}{\sum_l s_l(t - R_i)}.$$

F	P	MIRTO				TEXCP				TEXCPSP				TRUMP			
		ThrPut	Link1	Link2	Link3	ThrPut	Link1	Link2	Link3	ThrPut	Link1	Link2	Link3	ThrPut	Link1	Link2	Link3
	y_1^1		100 (50)	-	-		50 (50)	-	-		69 (50)	-	-		96 (50)	-	-
y^1	y_2^1	100 (108)	-	0 (33)	-	109 (108)	-	35 (33)	-	100 (108)	-	16 (33)	-	96 (72)	-	0 (22)	-
	y_3^1		-	-	0 (25)		-	-	24 (25)		-	-	15 (25)		-	-	0 (0)
	y_2^1		-	100 (33)	-		-	35 (33)	-		-	73 (33)	0		-	100 (33)	-
y^2	y_2^2	100 (108)	-	-	0 (25)	109 (108)	-	-	24 (25)	135 (108)	-	-	31 (25)	100 (75)	-	-	0 (10)
	y_3^2		0 (50)	-	-		50 (50)	-	-		31 (50)	-	-		0 (32)	-	-
	y_3^1		-	-	100 (25)		-	-	28 (25)		-	-	35 (25)		-	-	61 (25)
y^3	y_2^3	100 (83)	-	-	0 (25)	82 (83)	-	-	24 (25)	65 (83)	-	-	19 (25)	61 (58)	-	-	0 (10)
	y_3^3		-	0 (33)	-		-	30 (33)	-		-	11 (33)	-		-	0 (23)	-

TABLE I
SCENARIO FOR FLOWS WITH NO RATE LIMITATIONS: RATE IN MBPS FOR FIFO AND FQ (IN BRACKETS). LINK1=(3,5), LINK2=(6,5), LINK3=(8,5).

C. Numerical evaluation

The algorithms' evaluation is obtained by solving the system of ODEs for source rates and link queues derived in previous sections. We use the method of Runge-Kutta of the 4-th order to solve the problem. We employ the Abilene network (Figure 4) as topology for our analysis. All link capacities are set to C=100Mbps, queue limit is 10 packets, propagation delays is negligible, and packet size is set to 1500B. RTT is then given by transmission plus queuing delay. We consider a hot-spot scenario toward node 5 that has an aggregate incoming capacity of 300Mbps. Three sources, nodes 2, 6, and 10 send traffic to the same destination, node 5 (flow y^1, y^2, y^3 respectively). We suppose every flow can exploit three paths to route traffic toward the destination. As path cost, we chose the sum of the inverse of the link capacities ($\propto$ to packet transmission delays) and paths are ranked in increasing cost order: e.g. the best path of flow y^1 is y_1^1, its second best path is y_2^1 while y_3^1 is the worst among its three available paths. Link (3,5), (6,5), (8,5) are the bottlenecks and are indicated as link 1, 2, 3 respectively.

Fig. 1. Time evolution of the throughput of TEXCP and TEXCPSP with FIFO scheduling over link 2.

Fig. 2. Time evolution of the throughput of MIRTO, TEXCP and TRUMP with FIFO scheduling in the nodes: flow 1.

Fig. 3. Time evolution of the throughput of MIRTO, TEXCP and TRUMP with FQ scheduling in the nodes: flow 1.

Fig. 4. Abilene Network Topology [6].

We analyze two different scenarios: a first one where all flows have infinite backlog and a second where one flow out three (flow y^2) is access limited to 50Mbps. Results include calculations for scenarios with the two queue policies: FIFO with drop tail and FQ with DLQF. All parameters used for the presented numerical analysis are summarized in table III. They all have been set following the rules included in the original papers to achieve stability and fast convergence.

Table I and IV (left side) reports results for the scenario where all flows have infinite backlog and the queuing discipline is FIFO; results with FQ scheduling are also reported in brackets. Observe that TEXCP appears in two versions whether or not it implements the feature that gives priority to shortest paths (TEXCPSP).

MIRTO allocates 100Mbps to all flows and employs only shortest paths when scheduling is FIFO. This is the solution that would be obtained by the optimization (for max-min fairness) because it allocates to all flows the same rate while it minimizes the network cost.

TEXCP tends to use an amount of bandwidth close to the fair rate along each path. We can better see this in Fig.1 showing TEXCP and TEXCPSP over link 2. We observe TEXCP almost fairly shares the link capacity among all paths crossing the link (y_2^1,y_1^2 and y_3^3). On the contrary, TEXCPSP allocates bandwidth proportionally to paths' RTTs so that y_1^2 is privileged followed by y_2^1 and y_3^3.

MIRTO and TEXCPSP have similar performance while TRUMP gets much less throughput. TRUMP might employ, as MIRTO and TEXCPSP, only shortest paths at light load, but it under-utilizes them because every path cost (proportional to the path length) is weighted by a common w parameter. This means that TRUMP might prefer to scarcely use longer paths as they are weighted as very expensive by a w parameter tuned for attaining a certain utilization on the shortest path. In this example y^2 attains 100% of the path capacity (100Mbps) on its best paths, while the other two flows y^1 (=96Mbps, 96%) and y^3 (=61Mbps, 61%) are more constrained, even whether not really required.

The presence of FQ makes almost all algorithms performing the same, except TRUMP which gets much less throughput. Indeed the rate allocated over each path is never larger than the fair rate (the max-min fair rate). However this is a suboptimal allocation, because for the same throughput more resources are used with respect to FIFO. In Table IV we show that using FQ, MIRTO uses up to 50% more bandwidth for carrying the same throughput (from 600Mbps to 939Mbps). TEXCPSP and TRUMP have a similar loss of performance while TEXCP is less affected because, as already noticed, it already allocates the fair rate along the paths.

The solution obtained in presence of FQ assures fairness at link level, but it is not fair among users, e.g. flow y^3 is always penalized. So FQ is not any more globally fair when multiple routes are employed. This highlights that the problem of fairness among users cannot be restricted to per-flow rate sharing along a network path.

	Utilization	Throughput	Utilization	Throughput
	flow 2 non rate limited		flow 2 rate limited	
MIRTO	600 (939)	300 (299)	725 (815)	300 (291)
TEXCP	932 (939)	300 (299)	926 (915)	298 (288)
TEXCPSP	731 (939)	300 (299)	857 (915)	298 (288)
TRUMP	475 (554)	257 (205)		

TABLE IV
SUMMARY: NETWORK USAGE AND FLOW THROUGHPUT IN MBPS. FIFO AND FQ (IN BRACKETS).

Table II and IV (right side) summarize additional results for the scenario where flow y^2 is rate limited (see table IV for global statistics). In this scenario we see that resources freed by flow 2 are exploited by other flows that fill longer paths with consequent larger costs for the network. Notice also that in presence of FQ the loss of performance (compared to the employ of FIFO) is less pronounced than the previous scenario. In all cases MIRTO consumes less resources for the same throughput. The impact of FQ on more heterogeneous and realistic scenarios, in which sources have peak rates spanning over a larger range, is an open interesting issue left for future work.

As far as concerns convergence times, let us consider Fig.2 that depicts the time evolution of the rate of flow y^1 through the available paths, in presence of FIFO scheduling within the nodes. TRUMP and TEXCP converge fast, being equation based, although all parameters need to be tuned ad hoc for each particular scenario. Remember that both algorithms rely on precise ECN, feeding back information on the available bandwidth along every path. On the contrary MIRTO probes paths to discover available bandwidth and relies on simple ECN. Our experience on multiple scenarios (non reported for lack of space), says that there is no simple rule to tune TRUMP and TEXCP as they are very much dependent on the network setup. Fig.3 reports the time evolution of the rate of flow y^1 in presence of FQ schedulers. We notice that MIRTO suffers of higher convergence time with respect to the case of FIFO schedulers, while TEXCP and TRUMP convergence times are not affected.

VI. DISCUSSION AND CONCLUSIONS

In this paper we have introduced a novel routing algorithm, MIRTO, that is designed around some simple observations inferred from the water filling procedure used in the optimization problem of routing and flow control under max-min fairness.

We have presented a fluid model of MIRTO under FIFO and FQ at network links and compared it to other two multi-path routing algorithms, TEXCP ([17]) and TRUMP ([13]) modeled as well through ODEs. The comparison is made through numerical evaluations, for which we have reported the most significant scenarios.

As shown by the evaluation, MIRTO tends to use secondary paths only whether really needed. This is actually a behavior that mimics the water filling procedure of max-min that allocates rate on low cost paths first for all flows, and load balance

		MIRTO				TEXCP				TEXCPSP			
F	P	ThrPut	Link1	Link2	Link3	ThrPut	Link1	Link2	Link3	ThrPut	Link1	Link2	Link3
	y_1^1		100 (100)	-	-		80 (70)	-	-		88 (70)	-	-
y^1	y_2^1	125 (158)	-	25 (33)	-	151 (140)	-	45 (42)	-	158 (140)	-	40 (42)	-
	y_3^1		-	-	0 (25)		-	-	26 (28)		-	-	30 (28)
	y_1^2		-	50 (33)	-		-	13 (15)	-		-	37 (15)	-
y^2	y_2^2	50 (50)	-	-	0 (17)	50 (50)	-	-	17 (10)	50 (50)	-	-	3 (10)
	y_3^2		0 (0)	-	-		20 (25)	-	-		10 (25)	-	-
	y_1^3		-	-	100 (25)		-	-	31 (28)		-	-	23 (28)
y^3	y_2^3	125 (83)	-	-	0 (25)	97 (98)	-	-	26 (28)	87 (98)	-	-	43 (28)
	y_3^3		-	25 (33)	-		-	40 (42)	-		-	21 (42)	-

TABLE II
SUMMARY TABLE. FLOW 2 RATE LIMITED: RATE IN MBPS FOR FIFO AND FQ (IN BRACKETS).

MIRTO	TEXCP	TRUMP
$\alpha = 15kB$	$\alpha = 1$	$w = 0.1$
$\beta = 0.5$	$\beta = 1.41$	$\beta = 0.01$
$\gamma = 0.05$	$\gamma = 3$	$\gamma = 0.5$
-	$\epsilon = 10^{-3}$	-

TABLE III
PROTOCOLS' PARAMETERS

more rate on secondary paths only whether a flow have not yet attained its peak rate on primary paths. Setting parameters for MIRTO is fairly insensitive to network capacities.

TEXCP and TRUMP both have good performance in term of fast convergence if parameters are properly set, thanks to explicit congestion notification (also confirmed in a real protocol implementation evaluated in [27]). However TEXCP always makes use of longer paths, and it better performs with the shortest path bias (TEXCPSP). TRUMP's major fault is due to the fact that w (the weight of the network cost) is the same for all paths, limiting flexibility in networks with different link's capacities. Indeed the optimum w for high rate links is likely to be non optimal for low rate ones, resulting in unbalanced usage of network paths. However, even if a common w is not flexible, it allows for a more scalable architecture as it is set in every node for congestion notification, for every path to every flow.

We have also considered routing of single user's flows in networks deploying FIFO or FQ (or any other per-flow fair dropper like AFD) with very different results. Users employing multiple routes turns out to be suboptimal, whenever bandwidth allocation is forced to be fair at every link on a per-flow base. When sources have rate limitations the loss of performance due to FQ tends to vanish, especially for those sources with peak rates smaller than the fair rate. However a comprehensive analysis of this issue is left open for future work.

ACKNOWLEDGEMENT

This work has been partially funded by EU FP7 Integrated Project 4WARD (http://www.4ward-project.eu/). We thank Jim Roberts, Orange Labs, for the fruitful discussion on multi-path routing and Giovanna Carofiglio, Alcatel-Lucent Bell Labs France, for the discussion on fluid models and ODEs.

REFERENCES

[1] eMule, http://www.emule-project.net/
[2] Bit Torrent, http://www.bittorrent.com
[3] PPlive, http://www.pplive.com
[4] SOPCast, http://www.sopcast.com
[5] YouTube, http://www.youtube.com/
[6] Abilene Backbone. http://abilene.internet2.edu/.
[7] Briscoe, B; Jacquet , A; Di Cairano-Gilfedder, C; Salvatori, S. ; Soppera A; Koyabe M; Policing Congestion Response in an Internetwork Using Re-Feedback. in Proc. of ACM SIGCOMM 2005
[8] Carofiglio, G; Baccelli, F; Foss, S; Proxy Caching in Split TCP: Stability and Tail Asymptotics. in Proc. of IEEE INFOCOM 2008.
[9] Gallager, R. G.; A Minimum Delay Routing Algorithm Using Distributed Computation. IEEE Trans.on Communications, 25(1), Jan. 1977.
[10] Gallager, R. G. and Golestaani, S. J. Flow Control and Routing Algorithms for Data Networks. in Proc. of IEEE ICCC 1980.
[11] Golestaani, S.J.; A Unified Theory of Flow Control and Routing in Data Communication Networks Ph.D. Thesis, MIT Cambridge Lab for Information and decision systems, Dec. 1979.
[12] Han H.; Shakkottai S.; Hollot C.V.; Srikant R. and Towsley D.; Multi-path TCP: a joint congestion control and routing scheme to exploit path diversity in the internet. IEEE/ACM Trans. on Networking 14(6), Dec. 2006.
[13] He, J.; Suchara M.; Bresler, M.; Chiang, M. and Rexford, J. Rethinking Internet Traffic Management: From Multiple Decomposition to a Practical Approach, in Proc. of CONEXT 2007.
[14] Hahne,Ellen L.; Round Robin Scheduling for Fair Flow Control in Data Communication Networks. Ph.D. Thesis, MIT Cambridge Lab for Information and decision systems, December 1986.
[15] Hahne, E. L. and Gallager R. G.; Round robin scheduling for fair flow control in data communication networks, in Proc. of IEEE ICC 1986.
[16] Low, S.H.; Optimization flow control with on-line measurement or multiple paths, in Proc. of 16th International Teletraffic Congress,1999.
[17] Kandula, S.; Katabi, D.; Davie B.; Charnie A., Walking the tightrope: responsive yet stable traffic engineering. In proc.of ACM SIGCOMM 2005.
[18] Kelly, F.; Charging and rate control for elastic traffic. European Transactions on Telecommunications, vol. 8 1997.
[19] Kelly, F.; Maulloo, A. and Tan,D.; Rate control in communication networks: shadow prices, proportional fairness and stability. Journal of the Operational Research Society 49 (1998) 237-252.
[20] Kelly, F. and Voice, T.; Stability of end-to-end algorithms for joint routing and rate control. ACM SIGCOMM CCR 2005
[21] Key P. and Massoulié L.; Fluid models of integrated traffic and multipath routing Queueing Systems 2006 1:53 pp 85:98.
[22] Kortebi A.; Muscariello L.; Oueslati S. and Roberts J.; On the Scalability of Fair Queueing, in Proc. of ACM SIGCOMM HotNets III, 2004.
[23] Kortebi A., Muscariello L., Oueslati S., Roberts J.; Evaluating the number of active flows in a scheduler realizing fair statistical bandwidth sharing. in Proc. of ACM SIGMETRICS 2005.
[24] Misra, V.; Gong, W. and Towsley, D.; Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. in Proc. of ACM SIGCOMM 2000.
[25] Mo, J. and Walrand, J.; Fair end-to-end window-based congestion control. IEEE/ACM Transactions on Networking, 8(5):556–567 (2000).
[26] Muscariello, L.; Perino, D. and Rossi D.; Do Next Generation Networks need Path Diversity ? in Proc. of IEEE ICC, June 2009.
[27] Muscariello, L.; Perino, D.; Nardelli, B; Towards real implementations of dynamic robust routing exploiting path diversity. In Proc. of IEEE ICUMT 2009.
[28] Pan, R., Breslau, L., Prabhakar, B., and Shenker, S. 2003. Approximate fairness through differential dropping. SIGCOMM Comput. Commun. Rev. 33, 2 (Apr. 2003), 23-39.
[29] Srikant, R.; The Mathematics of Internet Congestion Control. Birk. 2004.
[30] Voice, T.; Stability of Multi-Path Dual Congestion Control Algorithms IEEE/ACM Transactions on Networking 15(6): 1231-1239, Dec. 2007