Modeling multi-path routing and congestion control under FIFO and Fair Queuing

Luca Muscariello[1] and Diego Perino[1,2]
[1]Orange Labs, France, first.last@orange-ftgroup.com - [2] INRIA Project Team "GANG", France

***Abstract*—Multi-path routing is a valuable on-line technique to deal with unpredictable and variable traffic patters, 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 analyze the performance of MIRTO, TEXCP and TRUMP, three recently proposed multi-path routing algorithms. Modeling of such algorithms is performed through fluid models, based on ordinary differential equations (ODEs). On a US-like backbone network, with and without in-network fair queuing schedulers, TEXCP and TRUMP show faster convergence times while MIRTO, that relies on simpler feedbacks, consumes less network resources.**

I. Introduction

Robust routing exploiting multiple paths is a powerful approach for overload control, mostly for intra-domain TE, inter-AS path selection under the same ISP, routing in wireless networks and metropolitan access networks.

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. 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 [6]).

In this paper we introduce an analytical model in order to compare different routing schemes using fluid ordinary differential equations (ODEs). The framework can be adapted according to the application, i.e. whether the protocol would route aggregates or flows. The sending rate of MIRTO [5], TEXCP [2] and TRUMP [3], is modeled and the performance of the three protocols is analyzed 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, and the modeling framework is a contribution in itself. TEXCP and TRUMP show faster convergence times than MIRTO, while this last protocol, that relies on simpler feedbacks, uses less network resources.

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

II. Fluid model of multi-path protocols

In this section, we model each source using a fluid representation of the sending rate $y_i^d(t)$ along each path i for a given flow d and we take delays into account. 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.

A. MIRTO

MIRTO [5] is a distributed algorithm built on the water-filling procedure of max-min fairness as basic criteria. The sending rate of every path is updated every T and the protocol starts to fill the best ranked paths first. All transmitters implement a window flow control protocol additive increase multiplicative decrease (AIMD) as TCP. Every egress node has to feed-back to the sources whether the paths are congested or not.

Since AIMD in presence of drop tail buffer management has not stationary solution, MIRTO agents average rate per path over a periodic cycle. If the average rate over a path varies with respect to its absolute value of only few percents, it is considered in "steady state". When all paths are in steady state MIRTO reduces the rate over all paths in order to prevent sub-optimal allocations.

Rates' evolution is described through a system of deterministic ODEs along the line of classical fluid models of TCP [4], [7]. 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:

$$\frac{dy_i^d(t)}{dt} = \frac{\alpha\psi_i^d(t)}{R_i^2} - \beta y_i^d(t)\phi_i^d(t-R_i) - \gamma \sum_{j\in\mathcal{P}_d} y_j^d(t)\zeta^d(t-R_i) \tag{1}$$

Let us illustrate each term in detail.

1) Increase term: the first term, at the right member of (1) 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 $\psi_i(t)$.

$$\psi_i^d(t) = \begin{cases} \prod_{j\in\mathcal{P}_d\setminus i} \mathbb{1}_{\{S_i^d < S_j^d\}} & \text{if } i \in \widetilde{\mathcal{P}}_d \\ \prod_{j\in\mathcal{P}_d\setminus\widetilde{\mathcal{P}}_d, j\neq i} \mathbb{1}_{\{S_i^d < S_j^d\}} & \text{if } i \in \mathcal{P}_d\setminus\widetilde{\mathcal{P}}_d. \end{cases} \tag{2}$$

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:

$$S_i^d(t) = \begin{cases} \sum_{k \in \mathcal{L}_i^d} \frac{1}{C_k} & \forall\, k \in \mathcal{L}_i^d \quad Q_k(t - R_i) < B_k \\ \infty & \exists\, k \in \mathcal{L}_i^d \quad Q_k(t - R_i) = B_k. \end{cases} \tag{3}$$

$Q_k(t)$ denotes the size of queue k at time t. As one can remark from (2), the decision function acts differently on paths that are in transitory or in steady state. The path selection described in 2, 3 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 $\mathcal{L}_i^d$ (3), i.e. the sum of the link costs.

2) Decrease terms: the second and the third term at the right member of (1) 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

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

regardless of the queue model. 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 $\zeta^d(t)$,

$$\zeta^d(t) = \prod_{j \in \mathcal{P}_d \setminus \widetilde{\mathcal{P}}_d} \mathbb{1}_{\{S_j^d(t) = \infty\}}$$

$\zeta^d(t)$ 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 $y_i^d(t) \wedge \left(\beta y_i^d(t) + \gamma \sum_{j \in \mathcal{P}_d} y_j^d(t)\right)$. We define a rate over a path to be in steady state 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 (1) 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.

B. TEXCP

TEXCP [2] is a distributed algorithm that balances load over multiple paths trying to minimize the maximum link load over the network. 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 [2] 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.

C. TRUMP

TRUMP is a network protocol that is obtained through decomposition of the optimization problem of routing and congestion control with network costs. 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 [3]. The resulting protocol requires nodes to evaluate the following link congestion measures:

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

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,

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

III. 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 ([1]).

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

$$\frac{dQ_k(t)}{dt} = A_k(t) - D_k(t) - L_k(t) \qquad (4)$$

where each term is defined according to

$$\begin{aligned} A_k(t) = &\quad \textstyle\sum_{d\in\mathcal{D}} r_k^d y_k^d(t) \\ D_k(t) = &\quad C_k \mathbb{1}_{\{Q_k(t)>0\}} \\ L_k(t) = &\quad (A_k(t) - C_k)^+ \mathbb{1}_{\{Q_k(t)=B_k\}} \end{aligned}$$

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

$$\frac{dQ_k^d(t)}{dt} = A_k^d(t) - D_k^d(t) - L_k^d(t), \quad \forall d \in \mathcal{D} \qquad (5)$$

where each term is defined according to

$$\begin{aligned} A_k^d(t) =& r_k^d y_k^d(t) \\ D_k^d(t) =& C_k \frac{\mathbb{1}_{\{Q_k^d(t)>0\}}}{\sum_{d':r_k^{d'}>0} \mathbb{1}_{\{Q_k^{d'}(t)>0\}}} \\ L_k^d(t) =& (A_k(t) - C_k)^+ \mathbb{1}_{\{d=\arg\max_{d'} Q_k^{d'}(t), \sum d' Q_k^{d'}(t) = B_k\}} \end{aligned}$$

$A^d(t)$ 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.

IV. Performance 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.

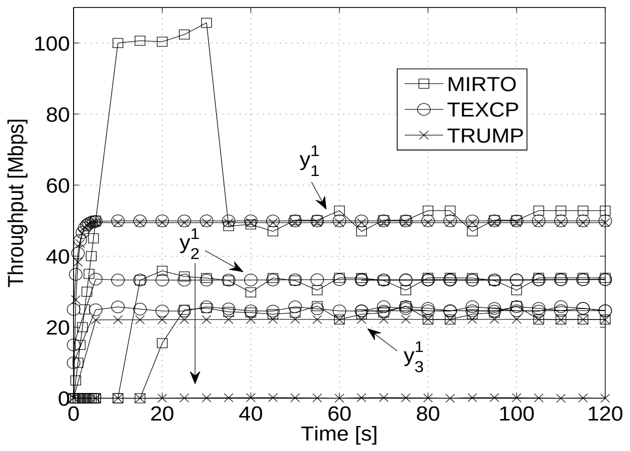

Fig. 1. Time evolution of the throughput of MIRTO, TEXCP and TRUMP with FQ scheduling in the nodes: flow 1.

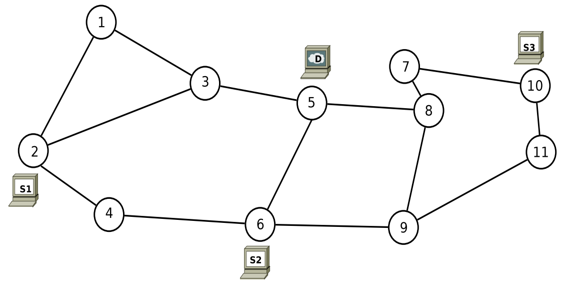

Fig. 2. Abilene Network Topology (http://abilene.internet2.edu/).

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

TABLE I
RATE IN MBPS FOR FIFO AND FQ (IN BRACKETS). LINK1=(3,5), LINK2=(6,5), LINK3=(8,5).

We employ the Abilene network (Figure 2) as topology for our analysis. All link capacities are set to C=100Mbps, queue limit is 10 packets, propagation delay 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. We consider a traffic scenario where all flows have infinite backlog and results include calculations for scenarios with the two queue policies: FIFO with drop tail and FQ with DLQF.

Table I and II reports the results (FQ in brackets). MIRTO allocates 100Mbps to all flows and employs only shortest paths when scheduling is FIFO. TEXCP tends to use an amount of bandwidth close to the fair rate along each path. On the contrary, TEXCPSP allocates bandwidth proportionally to paths' RTTs. 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.

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 II 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 allocates the fair rate along the paths.

As far as concern convergence times, let us consider Fig.1 that depicts the time evolution of the rate of flow y^1 through the available paths, in presence of FQ 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. In the case of FIFO scheduling, not reported here for lack of space, the convergence time of MIRTO decreases. TEXCP and TRUMP are not affected by the scheduling policy and they still converge faster than MIRTO.

	Utilization	Throughput
MIRTO	600 (939)	300 (299)
TEXCP	932 (939)	300 (299)
TEXCPSP	731 (939)	300 (299)
TRUMP	475 (554)	257 (205)

TABLE II
NETWORK USAGE AND FLOW THROUGHPUT IN MBPS. FIFO AND FQ (IN BRACKETS).

V. CONCLUSIONS

We have presented a fluid model of MIRTO, TEXCP and TRUMP under FIFO and FQ at network links and compared their performance by means numerical evaluations. TEXCP and TRUMP both have good performance in term of fast convergence thanks to explicit congestion notification. On the other hand, MIRTO uses less network resources and relies on simples feedbacks. The evaluation also highlighted that to force bandwidth allocation to be fair at every link on a per-flow base is sub-optimal.

REFERENCES

[1] 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.
[2] Kandula, S.; Katabi, D.; Davie B.; Charnie A., Walking the tightrope: responsive yet stable traffic engineering. In proc.of ACM SIGCOMM 2005.
[3] 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.
[4] Misra, V.; Gong, W. and Towsley, D.; Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. ACM SIGCOMM 2000.
[5] Muscariello,L and Perino,D; Early Experiences in Traffic Engineering Exploiting Path Diversity: A Practical Approach INRIA Technical Report No.6474, February 2008.
[6] 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.
[7] Srikant, R.; The Mathematics of Internet Congestion Control. Birk. 2004.