Towards real implementations of dynamic robust routing exploiting path diversity

Luca Muscariello[1], Diego Perino[1,2], Bruno Nardelli[3]
[1]Orange Labs Paris, France Télécom R&D
[2] INRIA Project Team "GANG", France, [3] Rice University, Houston, TX.
Email: firstname.secondname@orange-ftgroup.com, bruno.nardelli@rice.edu

Abstract—**In this paper we compare the performance of tree different dynamic traffic engineering algorithms exploiting path diversity in the Internet, TEXCP, TRUMP and MIRTO. We passed through a thorough implementation phase of these algorithms solving a number of issues related to protocol implementation that allows a complete analysis in real traffic settings in real networks. We discuss the performance of such protocols and some difficulties encountered during their implementation. Supported by an already mature standardization framework we conclude that dynamic yet stable traffic engineering is not only feasible but expected with rising interest by network operators.**

I. Introduction

Robust routing has many applications in today's networks as intra-domain traffic engineering (TE) and inter-AS path selection. In near the future, several key applications for a network operator, will also require robust routing. Some examples are service overlays and virtual networks that are rapidly gaining popularity thanks to their ease to build and deploy new services.

In the context of intra-domain TE, traditional issues are due to frequent line-card failures, large flow reroutes due to inconsistent routes' ranking coming from different routing protocols, exposure to highly variable, or unknown, traffic matrices. An additional challenge comes from access line upgrades that will attain, in the near future, capacities of 100Mbps. This will tempt every customer to run video applications (e.g. P2P-TV) or massive gaming (Internet video game tournaments), or other.

This upgrade process has just started and will exacerbate all the cited performance problems once such access lines will be dominant. Core network saturation is the chronicle of a death foretold that suggests deployments of robust overload control mechanisms.

Some recent propositions on traffic management as TEXCP [10] and then TRUMP [9], but also MIRTO [14]–[16] suggest some viable ways to optimise resource sharing and routing in today's networks. These solutions are supposed to be optimal with respect to a predefined objective. Typical objectives are to transport as many demands as possible minimizing the most loaded link or the global network cost. Such objectives allow to either reduce network upgrades or to limit the overall network congestion. Such objectives require that traffic sources be able to load balance traffic among multiple routes and to determine the optimal (or nearly optimal) rate over each of the available paths. Rate control plays an important role in order to keep stable routing decision.

In this paper we compare three routing algorithms that exploits path diversity: TEXCP [10], TRUMP [9] and MIRTO [14], [15]. We have implemented all these three algorithms and made intensive experimentation in a real test-bed over a network topology with the Abilene graph [1]. In this paper we show a number of significant scenarios in order to stress pros and cons of each algorithm with respect to their practical implementation and feasible deployments.

The paper is organized as follows: sec. II describes related work and sec. III the overall architecture. In sec. IV we provide a description of the considered routing protocols and in sec. V we describe the network set up. In sec. VI we compare the performance of these protocols in a network emulating the Abilene backbone. Sec. VII concludes the paper.

II. 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 [5]. 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 [6], [7]), that handle network rate limitations, using both path costs and flow rate adaptation. Kelly tackles the problem under a different perspective ([11], [12]), 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 [13], [21]. More recent works [8]–[10], [14], [15] have proposed a number of algorithms that may be implemented in real scenarios. Our contribution is related to verify the feasibility of the most performing algorithms in a real implementation and deployment in order to compare them in truly challenging scenarios.

III. Network architecture

We consider a network made of IP routers belonging to a network operator with some additional functionalities that can be deployed by MPLS. This network of routers supports path calculation among all origin destination pairs (OD pairs). Path calculation comprises multiple routes between two network endpoints. In this context endpoint are usually ingress and egress routers however in principle they may also be end to end users. This latter solution is far less scalable and would introduce a number of problems in terms of fairness ([15]) whether all users would not employ the same rate controller.

Routers are of two kinds, edge and core. In our context an edge router is notified a number of paths to reach egress nodes and every ingress node deploy a proxy that can route incoming traffic along multiple routes to reach the egress node, functioning as gateway towards the destination network. Core routers forward packets along the selected path. Traffic flows from ingress to egress nodes through multiple routes with adaptable rates with respect to path costs and network conditions. In this paper we assume sources employ homogeneous rate controllers.

A second fundamental requirement is that every router, for each outgoing network interface be able to notify congestion to the source, which may adapt sending rate accordingly. Each algorithm has more or less enhanced congestion notification mechanisms that are fed back to the transmitting nodes in different ways. For example a node experiencing congestion may directly notify link state through an explicit congestion notification packet (ECN) or may just signal congestion writing congestion measures on the headers of data packets flowing down to the egress node. In such a system downstream routers receive, from the upstream, congestion information written into packet headers and update it according to local congestion information before leaving packets flowing to the next hop. The egress node is then responsible for feeding back the congestion that has been elaborated along the whole path (see [4]).

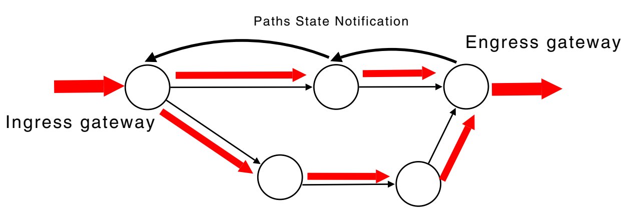


Fig. 1. Traffic is load balanced along different paths. Path state is measured and fed back to the source along the shortest path.

A. Control and Management Plane

The routing control plane computes multiple routes between all OD pairs and register a local path-tag to identify every OD-path initiated locally at every involved node. Packets are then tag-switched from the input to the output interface of the router with no additional processing. Traffic is encapsulated in PDUs and a new header added at the gateway with respect to the IP address of the egress node. The egress node is known according to shortest path routing, so that no additional processing is required. This new header carries some information as:

- path tags,
- congestion notification,

This system is similar to MPLS in particular in what concerns packet switching. Explicit congestion notification is currently not available in MPLS. However there is some interesting consensus on its useful properties as a very recent RFC [20] provides almost all that is needed to signal paths state. The set of protocols that we evaluate in sec. VI all have different ECN systems more or less complex. Therefore we had to develop a common flexible enough header to signal information. We will be back to this in more detail in sec. IV.

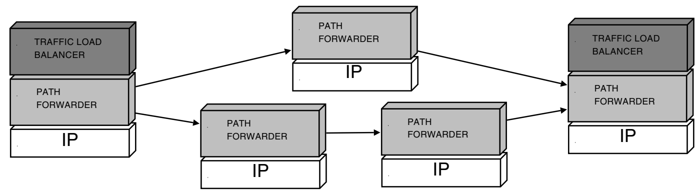


Fig. 2. Control and Management plane

IV. ROUTING ALGORITHMS

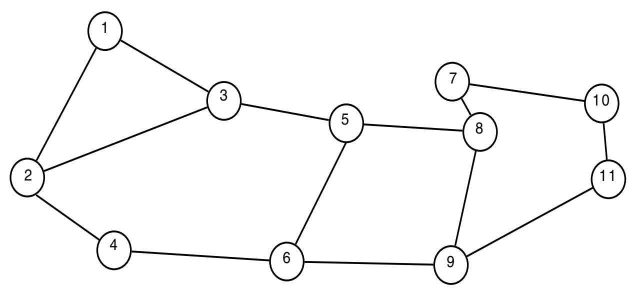


Fig. 3. Abilene Network Topology.

A. TEXCP

TEXCP is the first routing protocol that claimed dynamic traffic engineering be possible with good stability properties. This algorithm is described in [10] and performance are evaluated through simulations and some modeling. This algorithm takes into account links' utilization that is measured in every node and fed back to transmitters through periodic probes of period T_p and load balancing is adapted every $T_d \geq T_p$ with the following rule, for a source s on path p. The change on split ratios is then calculated:

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

where r_{sp} is the sending rate of 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 is used to weight rate allocations inversely proportional to the path length or delay, See [10] for more details. TEXCP has six parameters: α,β,γ,ϵ, T_p, T_d.

B. TRUMP

TRUMP is a network protocol that is obtain through decomposition of the optimization problem of routing with network cost. The objective to maximize is given by $\sum_s U_s - w \sum_l C_l$ 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 [9]. The resulting protocol requires nodes to evaluate the following link congestion measure

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

where N_T is the amount of bits arrive in the time interval T. s_l is then fed back to the source. 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 undertermined. Therefore we can only use TRUMP whether demands have no rate limitations. TRUMP has four parameters: w,β,γ, T.

C. MIRTO

The split ratio along available routes given by MIRTO is not equation based, i.e. there is no explicit expression that allows to calculate the rate to send along a route. MIRTO attains, for every path, a rate allocation which is inversely proportional to the path costs, following a dynamic *filling* procedure. A MIRTO rate controller update sending rates every T with respect to binary path state notifications (one bit ECN) received during this interval. ECNs signal whether the path has experienced congestion or not.

This information is used by the rate controller to adapt the sending rate with an AIMD paradigm. If the path is sensed congested at least once during the time interval T the path is marked as congested and a rate decrease of βy_p^s is applied to the sending rate of transmitter s on path p. If the path is never sensed congested over T the rate is increased of αy_p^s MIRTO requires a per path congestion notification like in [4] and in the MPLS framework like [20]. Core nodes do not require any other feature so that MIRTO perfectly fits to the standardisation boy of [20].

An additional requirement for MIRTO is that path be ranked with a static metric. It may be hop counts or a value proportional to maximum transmission time as $\sum_{l \in P} 1/C_l$ where the sum is made on all links along the path. We chose this latter to rank paths notified to the sender.

Hence, according to the algorithm 1, MIRTO fills the better ranked path first until either all the traffic demand is served or rate attains a stationary state. In this latter case more demand can be deflected to the second better ranked path. MIRTO iterates this procedure until either all traffic demand is served or all paths have been used. If all paths are used and the last path attains the stationary state, a rate decrease of $\gamma \sum_p y_p^s$ is applied over all path sending rate. MIRTO has four parameters: α,β,γ and T.

Algorithm 1 MIRTO: Every T

```
if ∃r_i^d : NotCongested( y_i^d ) and NotSteady(y_i^d) then
    for j ∈ P_d do
        if isCongested (y_j^d) then
            y_j^d = y_j^d − β y_j^d
        else if isSteady( y_j^d ) and isBestSteadyPath( y_j^d ) then
            y_j^d = y_j^d + α
        else if isNotSteady( y_j^d ) and isBestNonSteadyPath( y_j^d ) then
            y_j^d = y_j^d + α
        end if
    end for
else
    for j ∈ P_d do
        y_j^d = y_j^d − γ Σ_j y_j^d
    end for
end if
```

V. Implementation and deployment

In our testbed we have used a number of linux boxes (2 CPUs AMD Opteron 246, 2 GB of RAM) taken from the Grid 5000 platform [2], each connected to a switch through a 10Gbps line card. A series of IP tunnels are established to set up a chosen virtual network topology with link rate at 100Mbps. IP routing is deployed by the Quagga routing software suite [3] running OSPF. MPLS functionalities for path calculation and label switching is implemented in a software prototype including enhanced congestion calculation at nodes and notification to sources. Furthermore we have implemented rate control at ingress nodes supporting all the routing algorithms described in sec.IV. The ingress implements also a gateway for the upstream IP network whose traffic is transparently routed by our prototype exploiting path diversity within the network topology. At the egress a receiver reassembles traffic from multiple routes. We may also enable a reordering functionality of datagrams coming from multiple routes. This feature is not really necessary but, for sake of completeness in sec.VI-C we show some delay statistics that heavily impact reassembling performance.

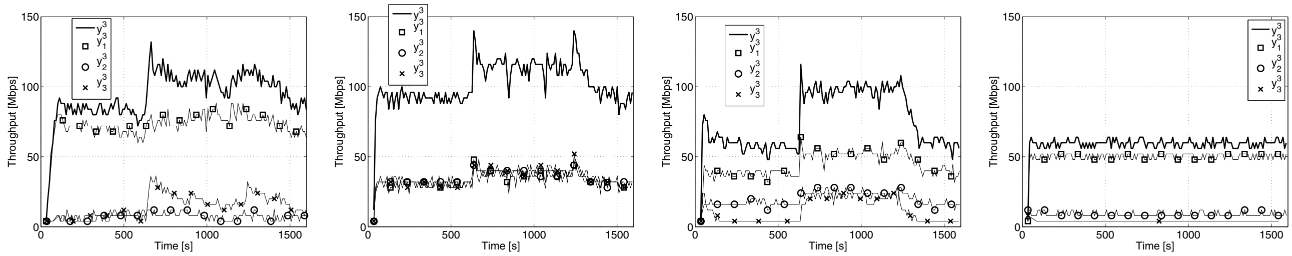
VI. Performance results

A. Parameter settings

Each of the protocols presented in sec.IV has different tuning rules for the parameters. We report here some remarks on this important aspect in practice on the base of a large set of experiments that we cannot report here for lack of

		MIRTO				TEXCP				TEXCPSP				TRUMP			
F	P	Throughput	Link1	Link2	Link3	Throughput	Link1	Link2	Link3	Thruput	Link1	Link2	Link3	Throughput	Link1	Link2	Link3
	y^1_1		73	-	-		31	-	-		64	-	-		69	-	-
y^1	y^1_2	95	-	16	-	91	-	31	-	88	-	5	-	77	-	4	-
	y^1_3		6	-	-		29	-	-		19	-	-		4	-	-
	y^1_2		-	69	-		-	29	-		-	85	-		-	92	-
y^2	y^2_2	90	-	-	12	90	-	-	30	129	-	-	36	100	-	-	4
	y^2_3		9	-	-		31	-	-		8	-	-		4	-	-
	y^1_3		-	-	67		-	-	31		-	-	39		-	-	51
y^3	y^3_2	85	-	-	8	93	-	-	31	59	-	-	16	64	-	-	9
	y^3_3		-	10	-		-	31	-		-	4.0	-		-	4	-

TABLE I
SUMMARY TABLE WHERE FLOWS HAVE NO RATE LIMITATIONS. STATISTICS IN MBIT/S. LINK1=(3,5), LINK2=(6,5), LINK3=(8,5).



(a) (b) (c) (d)

Fig. 4. Time evolution of the rate for MIRTO (a) TEXCP (b) TEXCPSP (c) TRUMP (d)

space. We have experimented scenarios with varying network capacities in order to have homogeneous and heterogeneous paths, and also considering different traffic matrices. Traffic matrices have been considered also variable in time in order to test the robustness of such algorithms to adapt to link failures or rapid changes in demands.

TEXCP seems to be easy to tune following the rule of thumb in the original paper, in all the considered scenarios. TEXCP with weighted rate allocation that takes into account path length or delay is stable only whether the metric does not vary in time. In homogeneous setting, where all links have the same capacity, hop count is a good metric that can be substituted by $\sum_{l\in P} 1/C_l$ for heterogeneous networks. The use of round trip time as path rank seems to be very difficult to tune and hardly ever work in every scenarios. Therefore we have chosen $\sum_{l\in P} 1/C_l$ to rank path as in MIRTO in the set of experiments that we present in sec.VI-C.

TRUMP is also easy to tune to obtain stability and rate convergence. The main problem with TRUMP is that the parameter w is difficult to chose in heterogeneous settings. w is responsible to weight the network cost with respect to the aggregated utilisation. w limits path usage for the overall network. However if some paths are heterogeneous that may result in greedy sources that saturate some paths and underutilise others. Moreover a good w parameter tuned to attain a certain load level over a one hop path (for a given demand) may result in very low path utilisation for other demands that have only longer paths. This would happen even whether those longer paths were lightly loaded. Therefore it is difficult to find a good w parameter for every scenario.

MIRTO has one bit ECNs therefore it suffers from all parameter setting issues of such AIMD senders. We have chosen parameter coming from the literature on TCP with good results. In tab. II we report tuned parameters for the scenario that we evaluate in VI-C.

MIRTO	TEXCP	TRUMP
$\alpha = 15kB$	$\alpha = 1$	$w = 0.1$
$\beta = 0.5$	$\beta = 1.41$	$\beta = 0.01$
$\gamma = 0.05$	$\gamma = 3$	$\gamma = 0.5$
$T = 1\text{sec}$	$\epsilon = 10^{-3}$	$T = 500\text{ms}$
-	$T_p = 500\text{ms}$	-
-	$T_d = 2.5\text{sec}$	-

TABLE II
PROTOCOLS' PARAMETERS

B. Experimental setting

We consider the Abilene network topology (Fig.3, [1]) that counts $N = 11$ nodes and set all link capacities to $C = 100$Mbps. We allow every source node to split traffic along its three best paths toward the destination node of a given flow.

We first analyse a small scenario in order to easily understand and explain the behaviour of the multi-path algorithms. We suppose there are 3 flows, y^1, y^2 and, y^3, generated by node 2,6 and 10 respectively, and directed to node 5. y^1 and y^3 have no rate limitations for all the experiment duration, while y^1 is limited from $t = 600$ to $t = 1200$sec, to a peak rate of 50Mbps. At $t = 1200$sec rate limitations are removed.

As TRUMP is designed to work with flows with no rate limitations we limit our analysis on TRUMP to a scenario where all flow have an unlimited peak rate for the whole experiment duration.

As a second set of experiments we consider two larger scenarios with more that 100 flows where we suppose they have a constant peak rate for all the experiment duration. Peak rate values follow a Log-Normal distribution with parameter $\mu = 16.6$ and $\sigma = 1.04$ (see [17], [18]). In these cases it is not possible analyse TRUMP behaviour because of flows with limited peak rate.

In a first scenario we generate a *uniform* traffic matrix. We suppose all nodes send traffic to all other nodes for a total of 110 flows in the network.

In a second scenario we generate a *hot-spot* traffic matrix. We assume every node generates 10 demands all directed toward the same destination node and aggregate them in a single flow. We suppose 70% of the traffic is directed to node 5 i.e. 8 nodes out of 11 send traffic to node 5. The remaining 30% is sent by 3 nodes to other 3 nodes other than node 5, for a total of 11 aggregated flows in the network.

C. Protocols comparison

We present here a set of performance measures in order to compare the considered algorithms. We start by considering the simplest scenario first, where only three flows are active. Starting from the case where all sources have no rate limitations in table I we see that MIRTO and TEXCP attain similar throughputs while MIRTO privileges shortest paths and TEXCP splits traffic almost equally among the available paths. TEXCPSP results to be unfair to the detriment of those flows who have a longer paths and TRUMP also which, moreover, attains smaller throughputs as the w parameter tends to limit network usage.

From the summary table IV we see that MIRTO employs less traffic to obtain similar throughput in this scenario. TEXCPSP has very similar aggregated results to MIRTO while, as we have already observed, fairness between flows is different.

If we limit the rate of demand 2 to 50Mbps we can observe in table III that results are very similar for all algorithms. MIRTO and TEXCPSP behave in a similar way while MIRTO tends to use preferentially shortest paths and then to consume less link capacity.

In fig.4 we show the time evolution of all algorithms in this scenario when a traffic flow (flow number 3) is limited to 50Mbps (excepting TRUMP) for a certain amount of time and how MIRTO, TEXCP and TEXCPSP react to new network conditions and adapt their rate accordingly. For this scenario we measure also the network delays along the paths a flow number 3 for all routing protocols. We report in fig. 5 the complementary distribution function of RTTs. TRUMP has low jitters as network delay varies a little with respect to the others. In addition delays are always smaller for TRUMP for path 1 and 2 while they are three time larger for the longest path. Delays in MIRTO are small and do not vary much, furthermore they have all the same distribution for all three paths. TEXCP and TEXCPSP also keep the same distribution for all path with larger delays with respect to MIRTO.

In this example MIRTO has shown to perform very well in terms of throughput, network utilisation and network delays.

For the two large scenarios we report in fig.6 the distribution of the link loads in the whole network for all the protocols plus shortest path routing with link weights given by 1/C. Of course min-cost routing makes use of less resources compared to multipath routing. For uniform traffic matrices utilisation is not very different to min-cost as all protocols tend to employ only one route except TEXCP which in any case prefers to split traffic in multiple routes even whether not needed. For hotspot traffic matrices MIRTO deviates a lot from min-cost and become closer to TEXCP as it starts using more secondary paths. In table V we report the aggregated information on throughput and utilisation showing how MIRTO performs very well in both cases.

		MIRTO				TEXCP				TEXCPSP			
F	P	Throughput	L1	L2	L3	Throughput	L1	L2	L3	Throughput	L1	L2	L3
	y_1^1		84	-	-		38	-	-		63	-	-
y^1	y_2^1	114	-	25	-	114	-	38	-	132	-	50	-
	y_3^1		5	-	-		38	-	-		19	-	-
	y_2^1		-	43	-		-	16	-		-	18	-
y^2	y_2^2	50	-	-	3	48	-	-	16	42	-	-	16
	y_3^2		4	-	-		16	-	-		8	-	-
	y_3^1		-	-	81		-	-	39		-	-	54
y^3	y_2^3	102	-	-	6	116	-	-	39	99	-	-	24
	y_3^3		-	15	-		-	38	-		-	21	-

TABLE III
SUMMARY TABLE WHERE FLOWS 2 IS RATE LIMITED TO 50MBIT/S. STATISTICS IN MBIT/S. LINK1=(3,5), LINK2=(6,5), LINK3=(8,5)

	Utilization	Throughput	Utilization	Throughput
	flow 2 non rate limited		flow 2 rate limited	
MIRTO	660	270	703	266
TEXCP	915	274	915	278
TEXCPSP	677	276	847	273
TRUMP	485	241	-	-

TABLE IV
SUMMARY: NETWORK USAGE AND FLOW THROUGHPUT THE SMALL SCENARIO. STATISTICS IN MBIT/S.

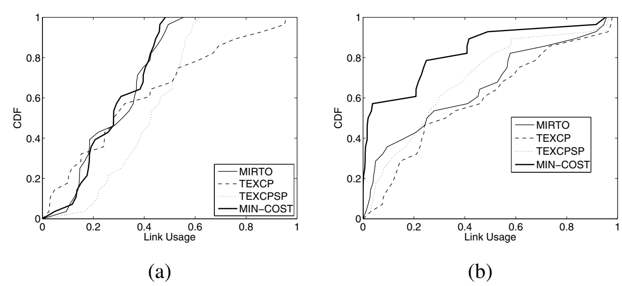
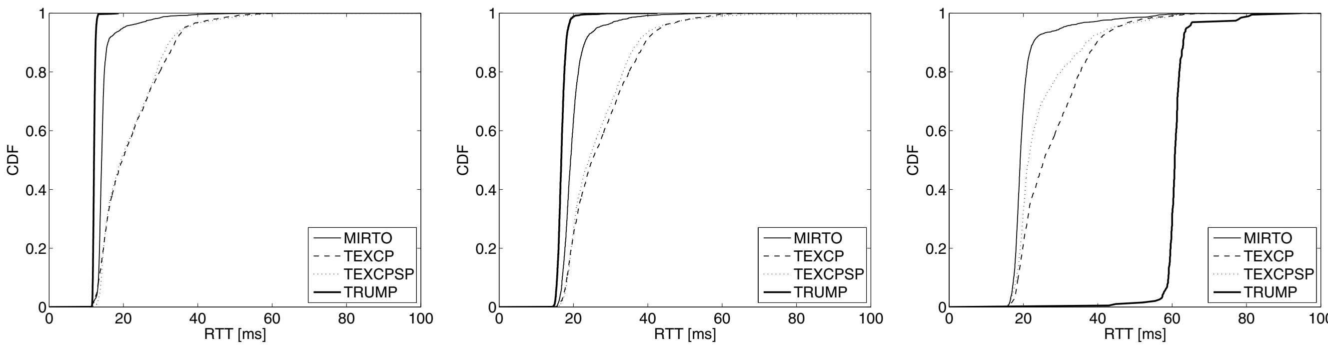


Fig. 6. Complementary distribution function of network link loads. Uniform traffic matrix (a) Hotspot traffic matrix (b)

	Utilization	Throughput	Utilization	Throughput
	Uniform		Hot spot	
MIRTO	826	311	1000	323
TEXCP	1180	308	1176	317
TEXCPSP	1168	310	904	301
MIN-COST	792	312	544	273

TABLE V
SUMMARY: NETWORK USAGE AND FLOW THROUGHPUT HOT-SPOT/UNIFORM SCENARIO. STATISTICS IN MBITS/S



(a) (b) (c)

Fig. 5. Cdf of network delays for the small scenario, for the flow number 3. Delay on path 1 (the shortest) (a) path 2 (intermediate length) (b) and path 3 (the longest one) (c).

VII. Discussion and Conclusions

In this paper we have considered multipath routing and rate control for optimality and robustness. We have considered three recently proposed routing protocols, TEXCP, TRUMP and MIRTO in a real implementation and deployment. Results are really encouraging as we have successfully deployed these protocols in realistic settings. This allowed to understand the complexity of tuning these protocols in multiple scenarios and their costs in terms of implementation. TEXCP is the first protocol proposed in the literature and seems to be agile and flexible enough for an implementation, while TRUMP has some limitations especially due to the w parameter which is common to all paths. This parameter comes out from the optimisation framework at the base of the TRUMP design where two opposite objectives are linearly combined: aggregated utility and network cost. This does not allow to satisfy correctly demands in heterogeneous scenarios as shown in the sec. VI. TEXCP waists lots of resource as it does not discriminate among shorter and longer paths. A modified version of TEXCP, TEXCPSP which privileges shortest paths may result in unfair rate allocation to flows constrained to use longer paths

MIRTO seems to perform better in terms of the measured performance parameters. However it has slightly slower convergence properties as it is not equation based as the other two. MIRTO moreover is simpler to implement in hardware as it does not rely on complex ECN but only on one bit congestion notification and may be deployed easily using the standardisation work of [20] while the other two would require much more efforts within forwarding nodes to compute ECN marks and signal them into packet headers.

Acknowledgement

This work has been partially funded by EU FP7 Integrated Project 4WARD. We thank Jim Roberts, Orange Labs, for the fruitful discussion on multi-path routing.

References

[1] Abilene Backbone. http://abilene.internet2.edu/.

[2] Grid5000 (G5K) Testbed Platform. https://www.grid5000.org.

[3] Quagga Routing Software Suite http://www.quagga.net/

[4] 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'05

[5] Gallager, R. G.; A Minimum Delay Routing Algorithm Using Distributed Computation. IEEE Trans.on Communications, 25(1), Jan. 1977.

[6] Gallager, R. G. and Golestaani, S. J. Flow Control and Routing Algorithms for Data Networks. In proc. of IEEE ICCC 1980.

[7] 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.

[8] Han H. , Shakkottai S. , Hollot C.V. , Srikant R. and Towsley D.; Multipath TCP: a joint congestion control and routing scheme to exploit path diversity in the internet. IEEE/ACM Transactions on Networking 14(6), December 2006.

[9] 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.

[10] Kandula, S.; Katabi, D.; Davie B.; Charnie A. Walking the tightrope: responsive yet stable traffic engineering. In proc. of ACM SIGCOMM 2005.

[11] Kelly, F.; Charging and rate control for elastic traffic. European Transactions on Telecommunications, vol. 8 1997.

[12] 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.

[13] Kelly, F. and Voice, T.; Stability of end-to-end algorithms for joint routing and rate control. ACM SIGCOMM CCR 2005

[14] Muscariello, L. and Perino, D Early Experiences in Traffic Engineering Exploiting Path Diversity: A Practical Approach. INRIA Technical Report No. 6474, 2008.

[15] Muscariello, L.; Perino; Modeling multi-path routing and congestion control under FIFO and Fair Queuing In Proc. of IEEE LCN 2009.

[16] Muscariello, L.; Perino; Evaluating the performance of multi-path routing and congestion control in presence of network resource management In Proc. of IEEE ICUMT 2009.

[17] Nucci A.; Sridharan A. and Taft N.; The problem of synthetically generating IP traffic matrices: initial recommendations, ACM SIGCOMM Computer Communication Review, 2005.

[18] Zhang,Y. ; Breslau,L. ; Paxson,V. ; Shenker,S.; "On the Characteristics and Origins of Internet Flow Rates" In proc. of ACM SIGCOMM Aug. 2002.

[19] M.Shayman, R. Jaeger, Internet Draft, Using ECN to Signal Congestion Within an MPLS Domain http://tools.ietf.org/html/draft-shayman-mpls-ecn-00

[20] Davie B., Briscoe B., Tay J. RFC5129, Explicit Congestion Marking in MPLS, January 2008, http://tools.ietf.org/html/rfc5129

[21] Voice, T.; Stability of Multi-Path Dual Congestion Control Algorithms IEEE/ACM Transactions on Networking 15(6): 1231-1239, Dec. 2007