On the Advantages of Cooperative and Social Smart Route Control

M. Yannuzzi, X. Masip-Bruin,
S. Sánchez-López, J. Domingo-Pascual
Advanced Broadband Communications Center
Technical University of Catalonia, Spain

A. Fonte[1, 2], M. Curado[1], and E. Monteiro[1]
[1] Laboratory of Communications and Telematics
University of Coimbra, Portugal
[2] Polytechnic Institute of Castelo Branco, Portugal

Abstract—**Smart route control is being increasingly used as a way to dynamically improve the end-to-end performance of the outbound traffic of multihomed stub domains. However, all the solutions available at present have in common two drawbacks which are key motivations for this work. First, all solutions are *standalone*, so no routing control interactions exist between the domains sourcing and sinking the traffic. The consequences of this lack of interactions are quite coarse route control over the outbound traffic from the domains, and the inability to smartly control how traffic flows into the domains. The second drawback is that all available solutions behave in a fully *selfish* way, that is, they operate without considering the effects of their decisions in the performance of the network. Given these limitations, we propose to extend the existing route control model from standalone and selfish to a *cooperative* and *social* route control model. Our main contribution in this paper is to show that when several route controllers compete for the same network resources, the conventional ones are outperformed by those using a *cooperative* and *social* approach and this becomes especially noticeable as the network utilization increases. Our results reveal that it is possible to reduce the frequency of traffic relocations by more than a 50% on average and still obtain slightly better end-to-end traffic performance for delay-sensitive applications. A key advantage is that our extensions can be installed and used today by simply performing software upgrades to any of the existing route control solutions.**

Keywords-component; Smart Route Control, end-to-end performance.

I. INTRODUCTION

Multihoming is a widespread practice exploited by stub Autonomous Systems (ASs), which consists of using multiple external links to connect to different Internet Service Providers (ISPs). By increasing their connectivity to the Internet, stub networks can potentially obtain several benefits, especially, in terms of resilience, cost, and traffic performance [1]. These are potential benefits since multihoming by itself is unable to guarantee the improvement of any of them. Thus, additional mechanisms are needed so as to accomplish such improvements. In particular, when an online mechanism actively controls how the traffic is distributed and routed among the different links connecting a stub network to the Internet, it is referred to as intelligent or smart route control.

At present, several manufacturers are developing and offering smart route control solutions targeting multihomed stub ASs [2-4]. All these solutions follow the same principle, that is, they actively improve the end-to-end performance of the traffic that flows *from* the AS toward a reduced set of "popular" destination prefixes[1].

This principle is supported by three facts. The first one is that in an AS, a small number of popular destination prefixes are responsible for a significant fraction of the interdomain traffic of the AS [5]. While this applies at the prefix-level, clearly, a correlation exists at the AS path-level so a similar conclusion can be draw. For instance, the measurements conducted in [6] reveal that only six AS paths carried about 36% of the one-month total traffic of a real multihomed stub AS.

The second fact is that popular destination prefixes represent remarkably stable entries in the BGP routing tables [5, 6]. And the third fact, is that actively tuning BGP so as to improve the performance of the traffic that flows *from* the AS is perfectly feasible, even, in short timescales. This is because the outbound traffic of an AS can be dynamically altered by means of BGP without needing to advertise the changes to the global Internet, i.e., without affecting any BGP router outside the local AS. Conversely, actively tuning BGP in order to improve the performance of the inbound traffic of an AS is unfeasible in rather short timescales. Controlling the flow of the inbound traffic of an AS implies to modify how upstream domains select their best path toward that AS. Unfortunately, this requires to advertise and propagate outside the AS every single tuning made in the AS, which normally affects the routing tables of a large fraction of BGP routers across the Internet. Clearly, it is not recommended to follow this approach too often. In addition, the effectiveness of controlling the inbound traffic of an AS is quite unpredictable, since it depends on the willingness of the upstream domains to honor the advertisements of the AS [7]. As a result, the existing concept of smart route control at the AS level, consists of improving (in short timescales) the end-to-end performance but for the outbound traffic only.

To accomplish such improvements, smart route controllers are capable of performing a series of tasks, which basically include discovery and monitoring of popular destination prefixes by means of passive and active measurements, and dynamic traffic relocation. In opposition to overlay networks [8], or interdomain tunnels [9], smart route controllers never circumvent BGP. Instead, they select on-the-fly the egress link from the AS for each popular destination prefix based on the outcome of their measurements.

This work was partially funded by the Spanish Ministry of Science and Technology under contract TSI2005-07520-C03-02, the Catalan Research Council under contract 2005-SGR00481, and the European Commission through CONTENT under contract FP6-0384239.

[1] A popular destination prefix is a prefix sinking a considerable amount of the traffic sourced from the AS. Its meaning is only local to each AS.

The effectiveness of this approach is confirmed not only by recent studies like [8], but also by the increased trend in the deployment of these solutions. Despite this, all the solutions available at present have in common two drawbacks which are key motivations for this work. First, all solutions are *standalone*, so no cooperation exists between the ASs sourcing and sinking the traffic (clearly no cooperation exists with the ASs providing transit to the traffic). The main consequences of this lack of cooperation are quite coarse route control over the outbound traffic of the ASs, and the inability to smartly control part of the inbound traffic.

The second drawback is that all available solutions behave in a fully *selfish* way, that is, they operate without considering the effects of their decisions in the performance of the network. Therefore, it becomes quite unclear to foresee if these route controllers could still perform so well if several of them compete for the same network resources. Conversely to a previous work which argues that the interference between multiple competing standalone route controllers causes only minor performance penalties [10], our work shows that in practice the penalties can be large, especially, when the network utilization increases. In that work, the performance penalty considered was the average latency, and it was evaluated at traffic equilibria[2]. Unfortunately, the available route control solutions at the AS-level are not precisely focused on seeking this kind of traffic equilibria. In addition, other performance penalties must be considered in practice, such as the implications associated with the number of traffic relocations needed to obtain a certain latency. For the route control solutions operating at the AS-level there are two major implications. First, each traffic relocation causes the flood of iBGP messages, so that *all* the BGP routers inside the AS learn about the new egress point from the AS to reach the popular destination. Second, it was recently found that bounces of traffic relocations and even oscillations may occur [11].

Given these limitations, we propose to extend the existing standalone and selfish route control model in two different ways. First, we propose that the route controllers belonging to a pair of multihomed stub ASs that exchange large amounts of traffic become capable of cooperating between each other. This cooperation will allow these ASs to improve the end-to-end performance of the traffic they exchange either in a one-way or a two-way fashion, depending on their specific needs and the way in which the majority of their traffic flows. An appealing advantage is that either of the two ASs can challenge the other to start the cooperation, which can be exploited by an AS so as to smartly control part of its inbound traffic, something which is unfeasible with the existing route control solutions. Second, we propose to endow each controller with a social route control algorithm, which adaptively restrains its intrinsic selfishness by learning from and evolving together with the network dynamics.

[2] Traffic equilibrium is the state in which no traffic can improve its performance by unilaterally changing its egress link assignment [10].

Our main contribution in this paper is to show that with these two extensions it is possible to outperform the existing standalone and selfish route control model, and the only thing actually needed is a software upgrade of the available route controllers. The figures 3, 4 and 5 support this claim. We have designed: i) a cooperative route control model; ii) the communication protocol between the route controllers; and iii) the social algorithm commanding the actions performed by the cooperative route controllers. Unfortunately, due to space limitations we cannot describe each of them in detail. Thus, our aim in this paper is to introduce the concepts of cooperation and social behavior in the smart route control area, and to show the most important results that we found.

II. Cooperative Route Control Model

An incentive for cooperation is to improve the way in which conventional route controllers monitor the network. In addition to passive measurements, *all* the route controllers available today, perform active probing for the most popular destination prefixes through all the available paths at the AS. A route controller constantly evaluates the end-to-end performance of these probes and selects the best path to route the traffic (and hence the best egress link), based on the lowest latency measured. Unfortunately, the current standalone controllers only consider the Round-Trip Time (RTT) latency metric, so they can only take coarse-grained routing decisions given that they decide how to route outbound without taking into account the Internet paths asymmetry. Furthermore, conventional route controllers perform the active measurements directly against the end-systems, so their success and precision actually depends on the willingness of the end-systems to accept and reply ICMP and TCP probes. In a cooperative framework, the route controllers can exploit the benefits of one-way measurements, such as One-Way Delays (OWDs), which can be performed directly between the route controllers, so the success and precision of the measurements becomes independent of the end-systems[3].

Another incentive for cooperation between domains is the potential benefits in terms of inbound traffic control. Standalone solutions are only able to control outbound traffic, which results appropriate for domains that are serving data to the greater Internet, but not for those which mainly receive data from the Internet. However, if both the source (S) and destination (D) domains could count with a Cooperative Route Controller (CRC), then the CRC in D could challenge the one in S to monitor and control the performance of the traffic flowing from S to D (see Fig. 1). The CRC in S could either accept or refuse to carry out such task depending on its own policies, its current load, and its particular needs.

With this in mind, we define "*cooperation between two distant domains*" as an association by which two peering CRCs

[3] We assume that the distant ASs sourcing and sinking the traffic are such that the difference between the exact end-to-end OWD of the traffic, and that measured between the route controllers belonging to the ASs is negligible.

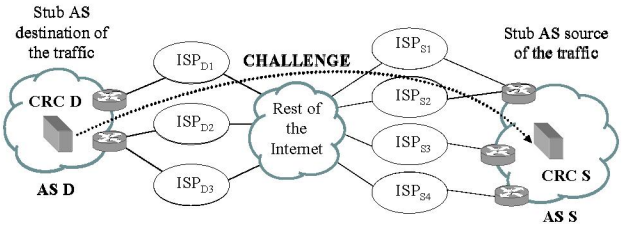

Fig.1. Cooperation between two distant CRCs (the transit ASs are not involved and remain unaware of this cooperation).

can agree upon a set of performance bounds, carry out one-way measurements, and exchange notification messages, either in a one-way or a two-way fashion. This cooperation is supported by a reliable communication protocol between peering CRCs, which we describe next.

DISCOVERY: A mechanism is needed to locate distant CRCs before the cooperation actually starts. An appealing option is to rely on the extensible nature of the DNS, and follow a similar approach to the one proposed by Bonaventure et al in [12]. With this approach, a new Resource Record (RR) called CONTROLLER can be added in the reverse DNS, as a pointer to a CRC. When a CRC wants to locate the CRC hosting a given prefix (either a popular source or destination prefix) it only needs to perform a reverse DNS lookup for the prefix, and ask for the CONTROLLER's address.

HANDSHAKING: Once the distant CRC is located a handshaking process starts. It should become clear that this process can be started by initiative of either the source AS or by the destination AS. The maximum tolerated one-way performance parameters of the traffic to be monitored and controlled are exchanged during this process. The outcome of a successful negotiation could be either that both CRCs are going to send probes to each other or that only one of them will do. This depends on the asymmetry of their traffic exchange, their local policies, and their particular needs. After this negotiation, both CRCs have synchronized their clocks so as to perform the one-way measurements [13].

KEEPALIVE: These messages are needed because the CRC sending the probes needs to be sure that the CRC receiving the probes is actually performing the OWD measurements and remains alive. Thus, the CRC receiving the probes is the one that sends the KEEPALIVEs.

MEASUREMENTS and NOTIFICATIONS: Once the initial negotiation has finished the communication between the CRCs in Fig.1 continues as follows (for simplicity we assume that the interest is just to monitor and control the traffic flow from S to D). S sends probes to D through all the available paths at S, just as conventional standalone route controllers do[4]. This means that D is receiving a set of probe flows (one for each available path at S, which is determined by the BGP routes available at S). D performs the one-way measurements and computes the median OWD for each of these flows. The motivation for choosing the median is that it is widely accepted as a very good estimator of the OWD that the user applications are actually experiencing.

If no performance changes are detected by D, i.e., *all* the medians remain unchanged, only KEEPALIVE messages are sent back from D to S (D does not probe S). It is worth highlighting that the median values are computed through a sliding window so as to leverage the notification reactivity of D. In case that any of the median changes or any other relevant event, D notifies S, so that the adaptive and social route control algorithm running on S can decide if the corresponding traffic needs to be reassigned or not to an alternative egress link of AS S.

III. Social Route Control Approach

Two candidate approaches can be adopted for the social route control algorithm running on S. On the one hand, the algorithm could follow a reactive approach, i.e., relocate traffic only when a pre-established bound is not fulfilled. An alternative is to follow a proactive approach by relocating traffic as soon as the performance becomes degraded up to some extent. After extensive evaluations, we have confirmed that controlled proactive approaches perform much better than the reactive ones. This claim applies not only in terms of end-to-end performance, but also in terms of performance penalties. The reason for this latter is that proactive approaches are able to anticipate network congestion situations, which in the reactive case, typically demand several traffic relocations when congestion has already been reached. Accordingly, the social route control algorithm in S uses a proactive approach.

In the sequel we provide a high-level description of the social route control algorithm that we have heuristically designed and tested. Our goal in this paper is mainly to make public our evaluation results, so for the reader who wants to get into the details of the algorithm please refer to [14].

Our heuristic is that S becomes capable of adaptively adjusting its proactivity depending on the network conditions. To be precise, S analyzes the evolution of the OWD using the notifications received from D, and depending on the OWD dynamics, S can adaptively restrain its traffic reassignments (i.e., its selfishness).

To accomplish this, S performs the following tasks. First, it classifies and groups the notifications received from D according to which particular flow of probes they belong to (recall that there is one flow of probes for each available path at S)[5]. From these groups of notifications, S obtains the evolution of the median OWD for each available path at S. The evolutions of these medians are precisely the inputs to our social route

[4] We highlight that we do not modify the measurement scheme used by conventional route controllers.

[5] This classification and grouping process is the exactly the same that conventional route controllers do. The only differences are that they probe directly against the end-systems, and that the "notifications" convey RTTs.

control algorithm.

Social Behavior of the algorithm: The social nature of the algorithm covers two different facets. On the one hand, the proactivity of S is controlled so as to avoid that minor changes in the medians trigger traffic relocations at S. The advantages of this approach are two-fold. First, it reduces the performance penalties associated with each traffic reassignment. And second, it avoids interfering too often with competing route controllers. For this reason, S filters the evolution of each of the medians. The filter works like an A/D converter. The outcome of this filter is what we call a Smoothed OWD (SOWD). An example of this filtering process for one of the available paths at S is depicted in Fig. 2. The social route control algorithm only takes into account and compares SOWDs. Thus, S may relocate certain traffic toward D *only* when a change in the SOWD along one of the available paths, produces a change in the best past selection at S (see Fig. 2). The number of paces that the SOWD needs to change so as to trigger a traffic reassignment can be configured by the administrator of S. We foresee hence different and configurable proactive strategies at the source CRC. The first advantage of this filter is that it produces the desired effect, that is, it prevents that minor changes in the medians trigger traffic relocations at S.

The second facet of the social behavior of the algorithm has to do with the dynamics of the median OWDs, to be precise, with how rapid are the variations in the evolution of the median values. The motivation for this is that when the median values start to show rather quick variations, the algorithm must react so as to avoid a large number of traffic reassignments in a short timescale. Such OWD dynamics typically occur when several smart route controllers compete for the same resources, leading to situations where their traffic reassignments interfere between each other.

To cope with this problem, our heuristic is to turn the filter in Fig. 2 into an adaptive filter. This filter is endowed with an adaptive pace of conversion, which is adjusted by the algorithm according to the evolution of the median OWDs. If the OWD conditions are smooth the pace is small, and more proactivity is allowed at S. However, if the OWD conditions may lead to instability the pace increases and the number of changes in the SOWD is diminished or even stopped until the network conditions become smooth once again. This has the effect of desynchronizing only the competing route controllers. Therefore, the second advantage of the filter is that it can be exploited by the source CRC to "socially" decide whether to reassign the traffic to an alternative egress link or not, and the degree of "sociability" of S is constantly adjusted by the adaptive nature of the filter.

IV. Evaluation Results

This section presents the simulation results performed to assess the advantages of the cooperative and social route control model. The performance of our CRCs is compared against the ones obtained with the following three alternative models:

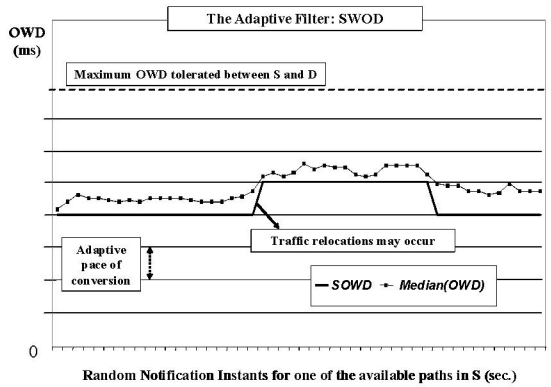

Fig.2. The adaptive filtering process exploited by the source CRC.

i) the conventional standalone and selfish model; ii) a cooperative model but without running the social algorithm; and iii) default BGP routing.

A. Evaluation Methodology

The simulation tests were carried out using the event-driven simulator J-Sim [15]. All the functionalities of the route controllers were developed on top of the BGP implementation available in this platform, i.e., the BGP Infonet suite.

AS-level Topology: For our simulation tests, the AS-level topology was built using the BRITE topology generator [16]. The topology was generated using the Waxman model with (α, β) set to (0.15, 0.2) [17], and it was composed of 100 ASs with a ratio of ASs to links of 1:3. This simulated network aims at representing an Internet core composed by ASs of ISPs able to provide connectivity and reachability to stub ASs. We assume that all ISPs operate PoPs (Points of Presence) through which the stub ASs can send and receive traffic. To emulate the stub ASs sourcing traffic toward popular destinations, we considered twelve ASs uniformly distributed across the AS-level topology. These stub ASs are connected to the routers located at the PoPs of three different ISPs. We considered triple-homed stub ASs because significant performance improvements are not expected from higher degrees of multihoming [1]. To emulate the stub ASs containing popular destinations we considered twenty-five ASs uniformly distributed across the AS-level topology. This gives an emulation of 12x25=300 pairs of CRCs competing for the same network resources during the simulations.

It is worth highlighting that the size of the AS-level topology used during our evaluations is small compared to the size of the Internet. However, to the best of our knowledge, this is the largest test made to assess the performance of different smart route control strategies in a competing environment.

Furthermore, given that smart route controllers operate in short timescales, we assumed that the AS-level topology remains invariant during the simulations.

Simulation Scenarios: In our experiments we run the same simulations separately using four different scenarios:

(i) Default defined BGP routing, i.e., BGP routers choose their best routes based on the shortest AS-path length.
(ii) BGP combined with the conventional standalone and selfish route control model at the stub domains.
(iii) BGP combined with a cooperative route control model at the stub domains, but without running the social algorithm.
(iv) BGP combined with a cooperative and social route control model at the stub domains.

For a more comprehensive comparison between the different models, we performed the simulations for three different network loads. We considered the following load scale factors (f): i) f = 1 (low load corresponding to 45% of the egress links capacity); ii) f = 1.5 (medium load corresponding to 67.5% of the egress links capacity); and iii) f = 2 (high load corresponding to 90% of the egress links capacity).

Synthetic Traffic and Simulation Conditions: The simulation tests were conducted using traffic aggregates sent from the source domains to each popular destination. These traffic aggregates are composed by a variable number of multiplexed Pareto flows, as a way to generate synthetic traffic demands, and hence to control the network load in the experiments. The flow arrivals are independently and uniformly distributed during the simulation runtime (i.e., the arrivals are described by a Poisson process). This approach aims at generating sufficient traffic variability so as to aid in the evaluations of the different route control strategies.

In addition, we used the following way to generate synthetic traffic demands for the remaining Internet traffic, called here background traffic[6]. We randomly pick four nodes. The first node acts as the origin (O) node of the traffic, and the remaining three nodes act as the destinations (D) of the traffic. We assigned one Pareto flow for each O-D pair. This process continues until all the nodes are assigned with three outgoing flows (including those in the multihomed stub ASs and those in the ISPs). The motivation behind this approach is to keep the overall average link utilization around the target of 10%, and to also contribute to the traffic variability. All background connections were active during the simulation runtime.

The frequency and size of the probes sent by the route controllers were correlated with the outbound traffic being controlled (just as conventional route controllers do [2-4]).

We assume that peering smart route controllers belonging to different stub ASs spanning several AS hops are able to exchange and agree upon a set of loose one-way performance bounds regarding the traffic between them. For instance, the International Telecommunication Union, Telecommunication Standardization Sector (ITU-T) G.114 suggests a OWD bound of 150 milliseconds to maintain high voice quality. Thus, for VoIP traffic we considered this bound as the maximum OWD tolerated for the simulated scenarios (iii) and (iv). In the standalone and selfish case, i.e. scenario (ii), the maximum RTT tolerated was chosen to be twice the OWD bound, namely, 300ms.

First Objective (Performance Penalties): The first objective of the simulation study is to demonstrate how the social nature of the CRCs contributes to reduce the performance penalties associated with frequent traffic relocations. To achieve this goal, we compared the average number of path shifts per second that occurred in the twelve competing stub ASs, for the scenarios (ii), (iii), and (iv). The number of path shifts is obtained by adding the number of path changes that are needed to meet the target OWD bound for each popular destination.

It is worth highlighting that in all the route control models assessed the route controllers operate independently and compete for the same network resources. Each CRC only cooperates with its remote peers. This allows us to assess the overall impact on the traffic caused by the interference between several smart route controllers running at different stub ASs. Thus, while analyzing the results for the different route control models, it is important to keep in mind that we will be taking into account all the competing route controllers present in the network.

Second Objective (end-to-end traffic performance): The second objective of the simulation study is to assess how the different route control models aid to improve the end-to-end traffic performance. To achieve this goal, we assessed both the outbound and the inbound traffic performance. For the outbound traffic, we compared the average OWDs obtained at the twelve competing stub ASs for all scenarios. The averages are computed at the stub domains taking into account the one-way latency to reach each of the twenty-five popular destinations. For the inbound traffic, the averages are computed at the stub ASs holding popular destinations, and taking into account the one-way latency from all the popular sources with peering CRCs.

B. Evaluation of Performance Penalties

Fig.3.a) illustrates the average frequency of path shifts performed in all the stub ASs for the three different load scale factors. Our results reveal that the cooperative and social route control model drastically reduces the frequency of path shifts compared to both the conventional model and a cooperative model without exploiting the strengths of the social route control algorithm. An important result is that the average reductions are significant for all the load scale factors assessed. When compared with the conventional route control model, the cooperative and social model contributes to reductions that vary between 50% for f = 2, up to 73% for f = 1.5.

Fig.3.b) allows us to observe the average frequency of path shifts on a per-domain basis. This is shown only for the highest load scale factor, i.e. f = 2, since the results obtained for the other two load scale factors are consistent with these, and hence they do not supply relevant information. The most important things to notice from Fig.3.b) are:

[6] These traffic demands aim at representing the traffic exchange between stub ASs not using smart routing.

i) When contrasting the conventional route control model against the cooperative and social route control model, *all* the competing ASs are able to reduce the frequency of their path shifts, and hence reduce the associated performance penalties.

ii) These reductions are indeed significant for all the stub ASs, except for AS10 which only achieves a marginal improvement.

In order to assess the effectiveness of the cooperative and social route control model, it is mandatory to confirm that the reductions obtained in the performance penalties are not excessive, so as to have a negative impact on end-to-end traffic performance. This is analyzed in the sequel.

C. Evaluation of end-to-end traffic performance

The results in Fig.4.a) show that the default defined BGP routing scheme has the worst average one-way latency for all the load scale factors considered. This was naturally expected, since it is a well known fact that the best paths chosen by BGP are usually not correlated with those paths exhibiting the best end-to-end performance for the users' traffic.

Fig.4.a) shows that the average OWD is drastically reduced when any of the smart route control solutions is used. The figure shows that the three route control models assessed show almost the same end-to-end performance when the network load is rather low (f = 1). When the network load gets higher (f = 1.5), the two cooperative route control models are able to improve the average OWD when compared with the conventional standalone and selfish route control model. The relative improvements against this latter are, 7.5% for the cooperative model without exploiting the social route control algorithm, and 12.5% for the cooperative and social model. For the highest load scale factor (f = 2) the cooperative models still perform better than the conventional route control model, but the relative improvements are less than for f = 1.5. The relative improvements are now 6% for the cooperative model without exploiting the social route control algorithm, and 4% for the cooperative and social model. The most important conclusion that can be extracted from these results is:

The cooperative and social route control model not only drastically reduces the performance penalties, but it also supplies slightly better end-to-end traffic performance for all the load scale factors assessed. This suggests that lots of the path shifts performed by the conventional route controllers are actually unnecessary in competing environments.

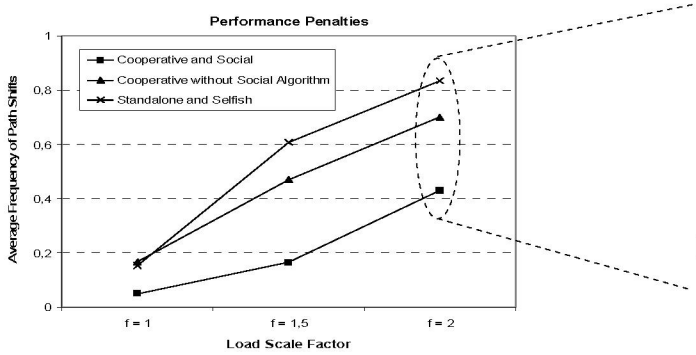

Fig.3.a) Average Frequency of path shifts for different load scale factors.

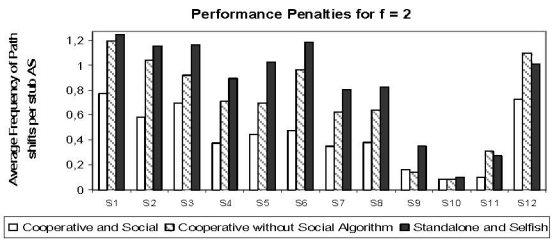

Fig.3.b) Average Frequency of path shifts per-stub AS for f = 2.

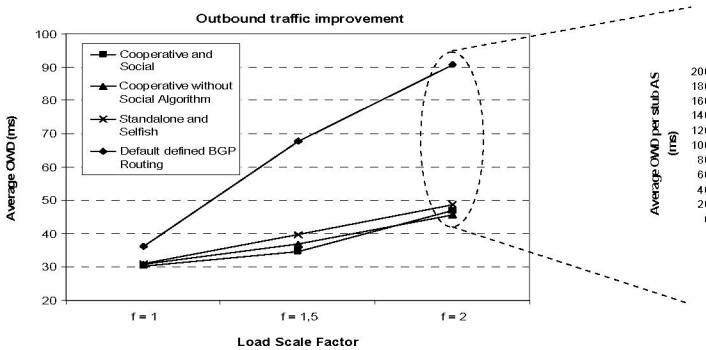

Fig.4.a) Average one-way latency for different load scale factors.

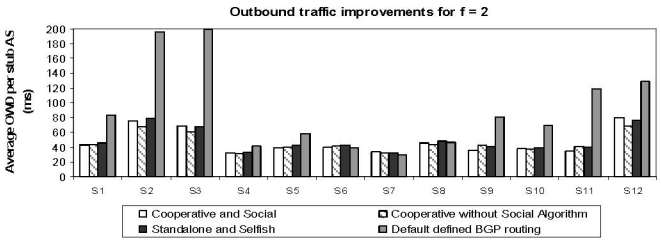

Fig.4.b) Average one-way latency per-stub AS for f = 2.

However, social improvements are usually not for free, and in our case this is confirmed in Fig.4.b). This figure allows us to observe the average OWDs obtained on a per-domain basis. Once again, this is only shown for the highest load scale factor, i.e. f = 2. The results obtained for the other two load scale factors are consistent with these, and hence they do not supply relevant information. Fig.4.b) shows that for two ASs, namely, AS7 and AS12 the cooperative and social model performs slightly worse than the conventional route control model. Nevertheless, the average OWD penalties are only about a few milliseconds, and Fig.3.b) reveals that both ASs achieve significant reductions in terms of path shifts.

Finally, we have compared the inbound traffic improvements relative to BGP routing. Fig. 5 shows the average OWD reductions obtained for the different load scale factors. Clearly, the conventional standalone and selfish route control model is not assessed in this case, since it cannot be exploited for inbound traffic control. The results in Fig. 5 reveal that the improvements in terms of one-way latency are really large, and as expected, the improvements are especially noticeable for higher load scale factors. The results in Fig. 5 were obtained when *all* the popular sources accepted the challenges from the destination domains. Thus, depending on the local policies of the source domains, the average improvements can be less than the ones shown here, especially when one or more sources start to reject the challenges.

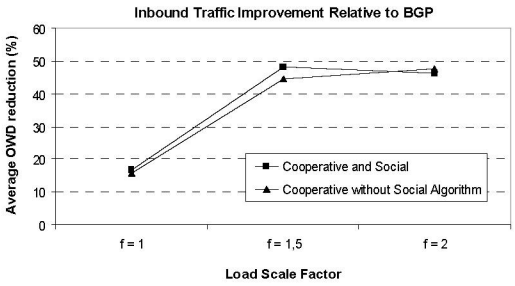

Fig. 5. Inbound traffic improvement for different load scale factors.

V. Conclusions

This paper has demonstrated that the current standalone and selfish route control model at the AS-level is far from optimal. Two simple extensions, such as the introduction of a route control protocol and a modified route control algorithm, are enough to outperform the existing route control model. These extensions can be incrementally introduced today as software upgrades, leveraging the cooperation and social behavior of the existing route controllers. Our first contribution has been to show that with these two extensions, the performance penalties can be drastically reduced on average and still obtain slightly better end-to-end traffic performance. Our second contribution has been to show the potential benefits of these extensions in terms of inbound traffic control.

It is important to highlight that the extensions proposed in this paper do not compromise the scalability of the current route controllers. The existing route control solutions are able to monitor and control more than 100 popular destination prefixes along each available path at the source domain [2-4].

Our extensions neither modify the core of the monitoring system nor intend to increase the scale of these solutions. On the contrary, our aim is to endow the route controllers with mechanisms that allow them to "socially" deal with their intrinsic selfishness.

Despite the abovementioned strengths, further research is needed. Even though our heuristic route control algorithm has proven to drastically reduce the performance penalties without impacting on the one-way latency, we cannot guarantee at this stage that this heuristic is the best possible approach. We hope that the results shown in this paper together with some other recent works like [11], lead to further research in this area.

References

[1] A. Akella, B. Maggs, S. Seshan, A. Shaikh, R. Sitaraman, "A Measurement-Based Analysis of Multihoming," in Proceed. of ACM SIGCOMM, Karlsruhe, Germany, 2003.

[2] Internap Networks, Inc., "Flow Control Platform".

[3] Avaya, Inc., "Converged Network Analyzer".

[4] Cisco Systems, Inc., "Optimized Edge Routing".

[5] J. Rexford, J. Wang, Z. Xiao, and Y. Zhang, "BGP Routing Stability of Popular Destinations," in Proceed. of the second ACM SIGCOMM Internet Measurement Workshop, pp. 197–202, November 2002.

[6] S. Uhlig, V. Magnin, O. Bonaventure, C. Rapier, L. Deri, "Implications of the Topological Properties of Internet Traffic on Traffic Engineering," in Proceed. of the 19th ACM Symposium on Applied Computing, Special Track on Computer Networks, Nicosia, Cyprus, March 2004.

[7] M. Yannuzzi, X. Masip-Bruin, and O. Bonaventure, "Open Isssues in Interdomain Routing: a survey," IEEE Network, Vol. 19, No. 6, November/December 2005.

[8] A. Akella, J. Pang, B. Maggs, S. Seshan and A. Shaikh, "A Comparison of Overlay Routing and Multihoming Route Control," in Proceed. of ACM SIGCOMM, Portland, USA, August 2004.

[9] B. Quoitin and O. Bonaventure, "A cooperative approach to inter-domain traffic engineering," in Proceed. of NGI 2005, Rome, Italy, April 2005.

[10] D. K. Goldenberg, L. Qiu, H. Xie, Y. R. Yang, and Y. Zhang, "Optimizing cost and performance for multihoming," in Proceed. of ACM SIGCOMM, August 2004.

[11] R. Gao, C. Dovrolis, E. W. Zegura, "Avoiding Oscillations due to Intelligent Route Control Systems," in Proceed. of INFOCOM 2006, Barcelona, Spain, April 2006.

[12] O. Bonaventure, C. de Launois, B. Quoitin, M. Yannuzzi, "Improving the quality of interdomain paths by using IP tunnels and the DNS," Technical Report, UCL, December 2004.

[13] F. Georgatos, F. Gruber, D. Karrenberg, M. Santcroos, A. Susanj, H. Uijterwaal, and R. Wilhelm, "Providing Active Measurements as a Regular Service for ISP's," in Proceed. of PAM, Amsterdam, Apr. 2001.

[14] M. Yannuzzi, A. Fonte, X. Masip, E. Monteiro, S. Sanchez, M. Curado, J. Domingo, "A Self-adaptive Interdomain Traffic Engineering Scheme," Technical Report UPC-DAC-RR-CBA-2005-8, December 2005. http://elio.ac.upc.edu/~dacsecre/reports/2005/64/RR-self-adaptive-interdomain-TE-scheme.pdf

[15] J-Sim Homepage, http://www.j-sim.org.

[16] A. Medina, A. Lakhina, I. Matta, J. Byers. "BRITE: An Approach to Universal Topology Generation", in Proceed. of MASCOTS, August 2001.

[17] B. Waxman. "Routing of Multipoint Connections," IEEE JSAC., December 1988.