Inter-domain Traffic Engineering using an AS-Level Multipath Routing Architecture

Pedro Amaral∗, Edgar Silva†, Luis Bernardo‡, Paulo Pinto§
Faculdade de Ciencias e Tecnologia, Universidade Nova de Lisboa,
Caparica Portugal Email: {pfa∗,ems†,lfb‡,pfp§}@fct.unl.pt

Abstract—Traffic Engineering (TE) aims at distributing traffic through routes to suit a specific goal. BGP addresses TE in a very specific way: it is a single path system, and mixes traffic control and route dissemination by letting attributes influence the choice of the best path. Several techniques have been proposed but due to BGP characteristics they have high complexity requirements such as the need for coordination and information about the network that is not locally available. and, in most cases, the use of tunnelling. In a previous work we proposed an inter-domain routing architecture DTIA (Dynamic Topological Information Architecture) that builds a robust multipath routing system based on the inter-AS relationships and routing policies. DTIA separates traffic control from route dissemination and seems to have less complex requirements to provide TE. The aim of this paper is to assess the ability of such a system to provide TE maintaining the complexity overhead to a minimum. We propose a simple feedback protocol to avoid congestion and achieve better traffic distribution at inter-domain level using only local available traffic information and DTIA's routing information.

Index Terms—Traffic Engineering; Protocols; Internet architecture; congestion control; policy routing.

I. INTRODUCTION

The Internet has evolved to a worldwide network that carries all kinds of traffic including mission critical and multimedia (e.g., video and voice over IP) and the need to control traffic is becoming greater than before. The reasons might be performance (choosing a low delay path or avoid congested network segments) or economical (using the cheapest provider).

Traffic engineering (TE) can be seen generally as the distribution of traffic to suit a specific goal. It has evolved with the need to efficiently use network resources and in close relation with the need to provide Quality of Service (QoS). At the routing level, where this paper is focused, it can be seen as a routing optimization problem. The complexity of this problem depends on a number of factors but they all lead to two main aspects: one is the amount of traffic control allowed by the routing protocol, which depends on protocol properties, like if it is multipath or single path, or if it allows the definition of end to end paths. The other is the amount of information available about the network conditions such as traffic, bandwidth or latencies.

The Internet is a large scale distributed network managed by several independent ASes that influence routing through their policies and the use of route attributes. This poses restrictions in both of the above aspects causing a small amount of traffic control and scarce network information.

We have been defining an architecture for inter-domain routing called Dynamic Topological Information Architecture, DTIA [1]–[3], that features a multipath routing environment without forwarding loops that allows traffic to be split arbitrarily amongst the different paths. In such an architecture a finer control over traffic is possible and traffic control can be achieved without changing routing configurations and without influencing route dissemination. This provides a new way to address the inter-domain TE problem: define the correct traffic distribution to achieve the TE goal instead of defining the correct route attributes that would lead to the traffic distribution.

In this paper we explore these DTIA’s characteristics to perform TE: multipath routing, and the ability to separate traffic control from route dissemination. Our intention is to assess what can be achieved by using them, keeping in mind that the Internet imposes limitations in terms of available information and AS coordination. Therefore, we follow a very lightweight approach to design a TE protocol that uses only DTIA characteristics, locally available information to each AS and minimal signalling consisting in a simple control packet. We then assess the amount of traffic control that is possible in such a constrained approach.

Our protocol uses DTIA’s routing and traffic information from the links directly connected to an initiating AS performing what is known as online TE. It uses requests for cooperation between the initiating AS and specific remote ASes. Cooperating ASes change their traffic distribution according to the feedback received from the initiating AS in order to avoid congestion in the network. The incentive to a cooperating AS is to increase the possibility that traffic leaving from it will avoid congested paths. The protocol acts similarly to a feedback mechanism that increases the cost of a path when one of its possible segments is becoming congested. We tested two different variations of the protocol in order to analyse the trade-off between performance and complexity.

The rest of the paper is organized as follows: In Section II we discuss TE concepts and identify the problems involved in achieving inter-domain TE. Section III presents some related work and a brief discussion in the proposed solutions for the problems. Section IV describes the DTIA architecture and the properties used in the TE protocol. Section V explores the possibility of performing lightweight TE using DTIA and

*This work was supported in part by the Portuguese Science and Technology Foundation (FCT/MCTES) through the PIDDAC Program funds (CTS multiannual funding), MPSat project, and the grant SFRH/BD/44476/2008.
defines the protocol. Section VI presents some experimental results. Finally the last section contains the conclusions.

II. INTER-DOMAIN TRAFFIC ENGINEERING

As we mentioned before, the complexity of the routing optimization problem depends essentially on the amount of traffic control provided by the routing platform in combination with the available network and traffic information. For the inter-domain scenario the amount of traffic control is usually small, the routing is based in a single path system (BGP) and current TE works mainly on the choice of a specific path for each destination with techniques based on manipulating the route decision process by setting the appropriate attributes in the announced routes, [4]. Moreover, information of network conditions can be hard to obtain due to the network scale and the distributed control. With such difficult control and scarce network information, inter-domain TE is usually limited to local objectives for each individual AS such as choosing the entry (inbound) and exit (outbound) points of traffic. In this context different ASes have different needs. For example, an AS hosting a popular content provider needs to control outbound traffic, while an AS having multiple application subscribers wants to control its inbound traffic. A transit domain moving traffic between ASes might need both. Although different in purpose the two types cannot be seen as independent since they impact the other.

Outbound traffic control is more easily managed than inbound. ASes control the decision process by tuning the border routers to use the appropriate attributes, usually using the LOCAL-PREF attribute. Also they can tune the IGP so that the proper border router is chosen for each destination [4]. However this control is always limited by the available routes received from neighbours since BGP is single path.

Inbound traffic control is a more complicated subject since it involves influencing the BGP decision process on other ASes. Several techniques are used [4] such as selective announcements, prefix de-aggregation, AS-Path prepending, MED, and redistribution communities. These techniques are rudimentary and pose a lot of constraints on the TE problem. There are side effects like the increase in the number of BGP advertisements caused by prefix de-aggregation that has an impact on the DFZ routing table size [5]. Also the interaction between attributes in BGP might not lead to the expected results. The optional community attribute is slightly different in nature and is also commonly used for TE purposes [6]. It works by overloading the meaning of the route announcements. The precise meaning is set between neighbours as well as the actions to perform. The main drawback [6] is the absence of a structured definition for the values and the lack of a standard method of advertising them to others.

The fact that traffic control is deeply connected to route dissemination has a major consequence in inter-domain TE: routing decisions and configurations are independent in each AS but changes in one AS affect the decisions in other ASes and can propagate in a cascade causing unpredicted instabilities. It is even worse when changes are performed without coordination and some knowledge of the network [7].

In the next section we present some related work in inter-domain TE and discuss the proposed solutions to the problem.

III. RELATED WORK

Inter-domain TE research has been focused mainly in providing enhanced control for the traditional outbound and inbound TE problems. Let us start with outbound TE research. Explicit routing using interdomain MPLS tunnels has been proposed to allow a finer control over outgoing traffic [8]. ASes can set the tunnels manually, based on the knowledge of the inter-domain links, or automatically based on traffic measurements. With this increased control several methodologies have been proposed that define local optimization problems and propose treatable heuristic algorithms to solve them. One example is [9] that proposes an algorithm to minimize expenses and maximize load balancing in stub ASes. More complex schemes, like MESCAL [10], aim at finding the optimal egress points that comply with a specific end-to-end bandwidth guarantee. MESCAL implies coordination between all ASes involved that define Service Level Agreements (SLAs) between them and the existence of traffic forecast matrices for offline engineering as well as a centralized routing server in the AS to deal with online TE. Lee et al. [11] describes an online approach for stub ASes that use traffic measurements as an input for egress selection.

Inbound TE research also exists. Wang et al. [12], for example, calculates the amount of prepending ASes for each prefix to achieve an objective but it needs to have information on maximum link loads and knowledge about the topology. In [13] a cooperative scheme using game theory and non-linear programming is used to solve a peering problem between two domains that agree to have a peering relationship. It determines how many peering points are needed and how ingress and egress points should be used by both ASes in order to maximize a given utility function. In this case signaling between the cooperating ASes is needed. In [14] inbound TE is performed by using IP tunneling to explicitly route between source and destination domains.

In summary, these solutions are mostly heuristic or try to mix optimization problems with heuristic algorithms to optimize simple AS inbound and outbound traffic. To deal with the limitations caused by the coupling between traffic control and routing dissemination there are two types of solutions (some approaches combine both): Tunneling and cooperation. Tunneling (using MPLS or other form) provides explicit end to end routes and traffic control becomes natural. However, tunneling requires cooperation and signalling. Also maintaining end to end inter-domain tunnels without affecting scalability in a network the size of the Internet is still an open problem. Cooperation between ASes uses network information like topology, traffic demands and current conditions to try to find the correct route manipulations so that changing the route dissemination does not cause instabilities. A trade-off between simplicity and optimality is obvious - better TE depends...
on better traffic control and this depends on new and more complex routing features, and on signalling and cooperation between ASes.

IV. THE DTIA ARCHITECTURE

DTIA is based on the current Internet business model and does not require any modifications to IP. It has a two-layer approach separating reachability (where path discovery is done) from routing (which performs path ranking). This separation also provides the possibility to control traffic without affecting the paths available at each AS and therefore without impacting the routing stability. Note that this provides a different paradigm to perform TE at inter-domain level. This section briefly describes the main features of DTIA having references for the details. We divided the description in control plane and forwarding plane, describing how the reachability and routing levels interact at each plane. We then present a useful DTIA property that concerns the possibility to infer remote routing information.

A. Control Plane

1) A Static AS level Map is used to obtain routes: The map describes the network based on the inter-relations amongst the ASes. It is static because it has no information about link failure. We justify our choice and possible deployment options for such a map in [1], [3]. The map is distributed via a flooding mechanism and a new version is distributed upon the addition of a new AS or link [3]. Failures are dealt by information messages that update the current state of the map and are only sent to the affected ASes [1], [3].

2) Reachability Level - Path Validation: The first step corresponds to the reachability. At this level routing policies are applied in the form of rules [1]. As a result an AS obtains all valid paths to all other ASes. In practical terms, this is similar to what would be available in BGP if all paths complying with the policies were announced and not just the best one. The difference is that the process is local to each AS and no routing messages are exchanged. Also, there can be more than one path to a destination providing a base for multipath routing, still maintaining the policies defined amongst the ASes. We conducted experiments and this process is quite feasible for networks with 12,000 ASes. In [2] we discussed how this process can scale to the entire Internet. Scaling is dependent on the time to calculate all the paths.

3) Routing Level - Path Ranking: DTIA’s routing protocol assigns a signature to each valid path. This signature is based on the business properties of the links forming the path (customer, provider, etc.). A preference relation is then defined over the signatures ranking the paths into sets. Paths belonging to the same set are equivalent, thus forming the multipath nature of the protocol. The relative preference between the sets is defined according to the routing policy. To prove certain consistencies and properties we used a routing algebra [15] defining the set of path signatures, how they are calculated, the preference relation, etc. A routing algebra \( A \) is a tuple \( A = (\Sigma, \prec, \oplus, L, \phi) \). \( \Sigma \) is a set of signatures that qualify the paths, \( \prec \) defines the preference relations between signatures, and \( \oplus \) is the operation to calculate the signature of a new path (see below). In [15] it is shown that if the operation has certain algebraic properties the protocol converges to a set of loop free paths with no forwarding loops.

DTIA considers the common policies used in BGP [16] plus two more to handle backup links and sibling relationships. Inter-AS links belong to one of four types (set \( L \)):

- **Provider-Customer** One AS (the provider) accepts all traffic from the other AS (the client). The link is labeled \( p2c \) in the provider-customer direction and \( c2p \) in the reverse;
- **Peer-to-peer** (\( p2p \)) ASes provide connectivity for their direct or indirect customers;
- **Peer-to-peer allowing backup** (\( p2pbk \)) the same as before but allows transit traffic if no other path exists;
- **Peer-to-peer allowing transit traffic** (\( p2patt \)) transit traffic is allowed in any situation (this models a sibling relationship).

Given these link types, the sets \( L \) and \( \Sigma \) are the following: \( L = \{p2patt, p2c, p2p, c2p, p2pbk\} \) and \( \Sigma = \{\varepsilon, P2Patt, P2C, P2P, P2Pbk, C2P, \phi\} \cup \{BKP \times N^+\} \). The preference relation for signatures (\( \prec \)) is listed below,

\[ P2Patt \approx P2C \prec P2P \approx P2Pbk \prec C2P \prec (BKP, 1) \prec (BKP, n), \]

where \( P2Patt \approx P2C \) are the most preferred signatures and \( (BKP, n) \) the least preferred one.

Table I shows the \( \oplus \) operation (the resultant signature when a link is added to a path in the direction to the origin).

<table>
<thead>
<tr>
<th>( \oplus )</th>
<th>( \varepsilon )</th>
<th>( P2Patt )</th>
<th>( P2C )</th>
<th>( P2P )</th>
<th>( P2Pbk )</th>
<th>( C2P )</th>
<th>( (BKP, x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p2patt )</td>
<td>( P2Patt )</td>
<td>( P2C )</td>
<td>( P2P )</td>
<td>( P2Pbk )</td>
<td>( C2P )</td>
<td>( (BKP, 1) )</td>
<td>( (BKP, x + 1) )</td>
</tr>
<tr>
<td>( p2c )</td>
<td>( P2C )</td>
<td>( P2C )</td>
<td>( P2P )</td>
<td>( \phi )</td>
<td>( (BKP, 1) )</td>
<td>( \phi )</td>
<td>( (BKP, x + 1) )</td>
</tr>
<tr>
<td>( p2p )</td>
<td>( P2P )</td>
<td>( P2P )</td>
<td>( P2P )</td>
<td>( \phi )</td>
<td>( (BKP, 1) )</td>
<td>( \phi )</td>
<td>( \phi )</td>
</tr>
<tr>
<td>( c2p )</td>
<td>( C2P )</td>
<td>( C2P )</td>
<td>( C2P )</td>
<td>( \phi )</td>
<td>( (BKP, 1) )</td>
<td>( (BKP, 1) )</td>
<td>( (BKP, x + 1) )</td>
</tr>
<tr>
<td>( p2pbk )</td>
<td>( P2Pbk )</td>
<td>( (BKP, 1) )</td>
<td>( P2Pbk )</td>
<td>( \phi )</td>
<td>( (BKP, 1) )</td>
<td>( (BKP, 1) )</td>
<td>( (BKP, x + 1) )</td>
</tr>
</tbody>
</table>

After ranking all paths, the final result of the routing layer is a forwarding table, based on AS numbers. This table can contain more than one path to a destination and paths are divided in sets according to the preference levels. The AS can then forward packets through any of the paths in the highest preferred set available. In [3] it is proven that no forwarding loops occur in these conditions. Routing loops are
not a problem either in DTIA because there are no routing messages in the control plane.

B. Forwarding Plane

Packets are forwarded using AS numbers instead of IP addresses, thus requiring a mapping service to map the destination IP address to the corresponding destination AS number. For details on the mapping service see [3].

C. Inferring remote routing information

All ASes have the AS level map of the network and rank the paths using the same $\oplus$ operation. If we consider a valid path $P$ from $X_1$ to $X_n$ as $P = X_1, X_2, X_3, \ldots, X_n$ and denote $L(X_i, X_j)$ as the label of the link in the $X_i$ to $X_j$ direction, then the signature is $S(P) = (((e \oplus L(X_{n-1}, X_n)) \oplus L(X_{n-2}, X_{n-1})) \oplus \ldots \oplus L(X_1, X_2))$. AS $X_1$ can apply $\oplus$ to the reverse path: $rP = X_n, X_{n-1}, X_3, X_2, X_1$ to calculate the signature of this path in the $X_n, X_1$ direction $S(rP) = (((e \oplus L(X_2, X_1)) \oplus L(X_3, X_2)) \oplus \ldots \oplus L(X_n, X_1))$. Generalizing, $X_1$ can calculate the signatures of all valid paths from any AS to it. In [3] we prove the following property:

Property 1: Given two ASes $A$ and $B$ there is no invalid path $P$ from $A$ to $B$ such that the reverse path $rP$ from $B$ to $A$ is valid.

So, $X_1$ can infer all paths that reach it from any given AS by reversing its valid paths.

V. DTIA Traffic Engineering Protocol

We have discussed in previous work [2], [3] how DTIA assumptions are in agreement with current practices and network characteristics and how it can be deployed. We also showed that it is scalable to Internet like topologies. The routing described so far is based on business policies. It provides multipath with routing stability and a mean to infer routing information of the other ASes. There is room now to perform TE by managing the traffic distribution by the available paths. DTIA intrinsically separates traffic control from route dissemination. The set of routes is stable and instead of trying to fine tune routes TE can be performed by finding the correct traffic distribution between the multiple paths. This is different from the traditional research since there is a stable set of routes without the need for tunnels or Label Switched Paths. Forwarding is still hop by hop and there is no global state on paths. Although this provides less control than tunnelling, it does not require signalling or state across ASes and is more resilient to failures. Taking advantage of this characteristics we focused on a network wide congestion avoidance algorithm based on link usage feedback. To maintain scalability and AS independence we use only local available traffic information and a unique feedback signalling packet.

The protocol works like a feedback mechanism that uses a signalling packet congestion feedback packet (CFP) that is sent by ASes that detect a congestion in one of their incoming links. It works by informing remote ASes about congested local links. Remote ASes can then redistribute traffic to other preferred paths which are more likely to avoid the congestion. The incentive to the remote AS is to forward its traffic through less congested paths, therefore reducing delay and packet loss. The redistribution of traffic will result in better traffic distribution depending on the number of cooperating ASes. The protocol has three phases: congestion detection, CFP distribution and traffic redistribution.

A. Congestion detection

Consider that one AS, $AS_{IN}$, has a set of incoming links $M$. The load on each link $m_i$ is monitored to see if a threshold value, $\psi_i$, is not passed. If the threshold is passed $AS_{IN}$ monitors the load per source-destination pair to identify the biggest contributor and sends a CFP packet to it.

B. CFP distribution

We actually implemented two different versions of the algorithm concerning the CFP packet distribution: pure feedback and selective feedback.

In the pure feedback version, $AS_{IN}$ identifies the largest contributor, $AS_{OS}$, and sends a CFP packet upstream that contains: the AS number of $AS_{OS}$, the AS number of the destination $AS_{Od}$, the AS number of the neighbour connected through $m_i$, denoted by $\nu_{mi}$, and the set of neighbours of $AS_{IN}$ that also route traffic from $AS_{Os}$ to $AS_{IN}$, which is denoted by $NO_{S_{IN}}$. $NO_{S_{IN}}$ contains the alternative ASes that can be used to route traffic from $AS_{OS}$ to $AS_{IN}$ avoiding the congested link.

When the AS $\nu_{mi}$ receives the CFP packet it checks if it has alternative paths to $AS_{IN}$ with the same signature and changes the traffic distribution to avoid the congested link. It then finds its own set of neighbours that also route traffic from $AS_{OS}$ ($NO_{S_{\nu_{mi}}}$), and a CFP is sent to ASes in $NO_{S_{\nu_{mi}}}$. The process is repeated by every AS that receives a CFP until the $AS_{OS}$ is reached. The simple topology in Figure 1 serves as an example. Consider AS C is the $AS_{IN}$ and that at a given time $t$ the link C-D becomes congested and the flow with more packets is G-C. A CFP packet is sent to AS D containing $AS_{Os} = G$, $AS_{Od} = C$, $\nu_{mi} = D$ and $NO_{i} = F, E$. Since AS D has two paths of C2P signature to C it changes the traffic distribution sending more traffic through AS F. It then sends CFD packets to $NO_{S_{D}}$ (the set of neighbours sending traffic from origin G) that contains only AS G. Since G is the origin, no more CFDs are sent. G checks if it can reach C through more than one path, and in this particular case it increases the amount of traffic through the G-E-C path stopping the process.

Fig. 1. Simple Topology
In the selective feedback version, we use the possibility to infer the routing information of $AS_{Ox}$ (by computing the signatures of the reverse paths) to send the CFP packet directly to the source of traffic. This allows us to infer the paths from $AS_{Ox}$ to $AS_{IN}$.

Let $F$ be the set of first hops in those paths. For every member $f_i$, we can know, by applying the same process, the paths from $f_i$ to $AS_{IN}$. We can then repeat the process until we reach $AS_{IN}$. A CFP packet is sent to all $f_i$ ASes that have alternative paths for the destination $AS_{Od}$. This uses DTIA remote routing inference capabilities to try to reduce the number of CFP packets. We use Figure 1 again as an example with the link C-D congested due to the flow G-C. In this case AS C will infer the paths from AS G to itself. There are three paths with signature C2P (G-D-C-G-D-F-C and G-E-F) where the first two are reduced to one since the first hop is the same (AS D). So a CFP packet is sent to AS G. AS C then checks the paths from D and E to itself (D and E are the first hops of paths from G to C). AS E has no alternatives and therefore a CFD message is not sent from C to E and this part of the process stops. AS D has alternatives and therefore a CFD packet is sent to it by C. Next, D first hops to C (F and C) are evaluated. Since C is the destination only F is checked for paths. AS F has no alternative paths and therefore CFP packets are not sent. Since F is directly connected to C the algorithm stops.

For both versions, when the link is no longer saturated and the total load from $AS_{Ox}$ decreases a RESET packet is sent to return to the previous default traffic distribution.

C. Traffic redistribution

Upon the reception of a CFP packet an AS is informed that a congestion is occurring in the link $m_i$ between $AS_{IN}$ and $ν_{mi}$, and that the flow with destination $AS_{Od}$ is the higher contributor to the congestion. To evaluate if it has alternative paths it checks the forwarding table, both for $AS_{Od}$ and $ν_{mi}$. If it finds first hops that only reach $AS_{Od}$ and not $ν_{mi}$ it knows that those are capable of reaching $AS_{Od}$ avoiding the congested link. Even paths to $AS_{Od}$ and $ν_{mi}$ starting with the same first hop can diverge along the way and also avoid link $m_i$. However, since DTIA has no explicit end-to-end paths, the concrete path followed after the first hop depends on the routing decisions of the other ASes. Therefore, avoiding $m_i$ is not guaranteed in this case. The distribution of traffic is then performed by assigning a higher weight to paths that reach both $AS_{Od}$ and $ν_{mi}$ and a smaller weight to the ones that only reach $AS_{Od}$. We used a simple exponential weight based traffic distribution scheme based on [17] to split the traffic. Considering the multiple paths between $i$ and $j$ indexed by $k$ the traffic distribution ratio is $X_{k}^{(i,j)}/X^{(i,j)}$ where $X_{k}^{(i,j)}$ is the amount of traffic in path $k$ and $X^{(i,j)}$ the total traffic through all paths. The distribution for path $k$ is given by:

$$\frac{e^{-w_{k}^{(i,j)}}}{\sum_{n}e^{-w_{n}^{(i,j)}}}$$

with $n$ being the $n$ paths available and $w_{n}^{(i,j)}$ the weight of the path $i$.

VI. EXPERIMENTS

In previous work we built a DTIA emulator to validate the reachability and routing level scalability [2], [3]. The results show that given the current state of the art in computational power DTIA scales nicely up to Internet like topologies with 11,335 ASes (the European region). In [2] we show how it can scale to the entire Internet. To test the TE protocol we need to perform packet level simulation, which is computationally more demanding. We implemented DTIA and the proposed TE protocol in the ns-2 simulator [18] in both the pure and selective feedback versions. As opposed to the DTIA routing, the TE protocol acts in a subset of the network affecting only the ASes forwarding traffic through the congested link. Therefore scale is not a primary issue. We opted then to use a small subset of the Internet to measure more easily the impact of the TE protocol in traffic distribution. We used a topology from the CAIDA AS Relationships Data research project [19] trimmed to 54 ASes and 517 links. This topology contains a subset of stub ASes from Portugal, and the set of transit ASes that they use, up to tier-1. It includes ten tier-1 ASes (without providers) at the top, and a set of lower tier transit ASes connected by $p \geq c$ links. The topology was then verified using the RIPE [20] database to be as close to the real network as possible.

In the current Internet a small set of ASes are the sources of a large percentage of the traffic. These ASes tend to be placed near the tier-1 transit ASes [21]. Traffic consumers are usually part of stub ASes that are less connected. Taking this in consideration we define sets of possible traffic senders and receivers from which we randomly choose three traffic sources and one destination. Initially all ASes distribute an equal amount of traffic through all paths of the highest preference set. The traffic sources are set so that the traffic capacity of links is exceeded. We experimented with and without the TE protocol running and used 110 different sets of traffic origins and destinations. We measured the amount of packets dropped in every congested link $m_i$. Figure 2 shows the cumulative distribution function (CDF) of the reduction in % of the number of drops for the 110 experiments. It shows the results
cases we have a reduction of at least 90% in the number of dropped packets in the network when the TE protocol is acting. The results are very similar for the two feedback mechanisms with the selective feedback getting marginally better performance. In the pure feedback case only 30% of the tests had a reduction lower than 50% and in the selective case only 20% were below that. Another important aspect to measure was the traffic distribution before and after the use of the TE protocol. We started by measuring the distribution of traffic between the M incoming links of AS\(I_{N}\) (including the congested link). To measure the distribution evenness we used the Jain coefficient given by \(\frac{\left(\sum_{i=1}^{n} m_i \right)^2}{\left(\sum_{i=1}^{n} m_i^2 \right)}\) where \(n\) is the total number of links in \(M\). The coefficient has the value 1 when all links receive the same amount of traffic. Figure 3 shows the CDF of the Jain coefficient for all congested link occurrences, measured for the 110 experiments. We can see that for 80% of the experiments the coefficient is below 0.5 without using the TE protocol. Using the TE protocol we obtained for 60% of the experiments a coefficient over 0.5. The numbers here are also very similar between the pure feedback and the selective feedback versions. These numbers show that if all ASes cooperate upon the receipt of a CFP packet we obtain significant results in congestion reduction and better traffic distribution, with a very low cost in terms of complexity on top of DTIA. This shows that DTIA can form a good base for TE protocols. In our case the major complexity added is the CFP packet distribution. We measured the total number of packets distributed for both the selective and pure feedback versions. The selective feedback version is an optimization to reduce the number of packets sent. The reduction of the average number dropped from 2551 packets in the 110 experiments to 1012. This reduction of 61.4% is the most significant difference between the two versions.

VII. CONCLUSIONS AND FURTHER WORK

Traffic Engineering at the inter-domain level is still a challenging area. Several approaches have been proposed and they usually have considerable complexity costs in the current Internet architecture making them far from being bases to solvable optimization problems. Hard guarantees in the Internet bring unimplementable complexity and it is likely that it will never happen with the current protocol suite. We opted for soft guarantees and took advantage of the characteristics of DTIA in terms of separation of concerns and multipath. This paper contains an evaluation of the complexity incurred. We added a simple signalling packet and used only local link information and DTIA features to avoid congestion and achieve better traffic distribution on top of DTIA. It is a minor cost especially because DTIA needs no packets (not even the route advertisements of BGP). The results were encouraging with significant gains to regular DTIA and could be applied to any other architecture that separates reachability and routing, and features multipath. An interesting topic for future work could be the definition of the necessary features to add to DTIA so that it can provide a routing base for solvable formal route optimization problems.

REFERENCES