Computer Networks 53 (2009) 60–80

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Inter-domain policy violations in multi-hop overlay routes: Analysis and mitigation ☆,☆☆

Srinivasan Seetharaman [a,*], Mostafa Ammar [b]

[a] Deutsche Telekom Laboratories, Los Altos, CA 94022, United States
[b] Networking and Telecommunications Group, College of Computing, Georgia Institute of Technology, 266 Ferst Drive, Atlanta, GA 30332, United States

ARTICLE INFO

Article history:
Received 24 October 2007
Received in revised form 31 March 2008
Accepted 17 September 2008
Available online 1 October 2008

Responsible Editor: Dr. A. Orda

Keywords:
Overlay routing
Inter-domain policies
Violations
Cross-layer conflict
ISP filtering

ABSTRACT

The Internet is a complex structure arising from the interconnection of numerous autonomous systems (AS), each exercising its own administrative policies to reflect the commercial agreements behind the interconnection. However, routing in service overlay networks is quite capable of violating these policies to its advantage. To prevent these violations, we see an impending drive in the current Internet to detect and filter overlay traffic. In this paper, we first present results from a case study overlay network, constructed on top of PlanetLab, that helps us gain insights into the frequency and characteristics of the different inter-domain policy violations. Further, we investigate the impact of two types of overlay traffic filtering that aim to prevent these routing policy violations: *blind filtering* and *policy-aware filtering*. We show that such filtering can be detrimental to the performance of overlay routing. We next consider two approaches that allow the overlay network to realize the full advantage of overlay routing in this context. In the first approach, overlay nodes are added so that good overlay paths do not represent inter-domain policy violations. In the second approach, the overlay acquires permits from certain ASes that allow certain policy violations to occur. We develop a single *cost-sharing* framework that allows the incorporation of both approaches into a single strategy. We formulate and solve an optimization problem that aims to determine how the overlay network should allocate a given budget between paying for additional overlay nodes and paying for permits (transit and exit) to ASes. We illustrate the use of this approach on our case study overlay network and evaluate its performance under varying network characteristics.

1. Introduction

Overlay networks have recently gained attention as a viable alternative to overcome functionality limitations (e.g., lack of QoS) of the Internet, or to enhance the services current being offered (e.g., rerouting around failures). The basic idea of overlay networks is to form a virtual network on top of the physical network so that overlay nodes can be customized to incorporate complex functionality without modifying the native IP network.[1] Typically, these overlays route packets over paths made up of one or more overlay links to achieve a specific end-to-end objective. Each intermediate overlay node is referred to as a *relay* and the forwarding operation at each hop is referred to as *relaying*.

The underlying Internet, on the other hand, is a complex structure arising from the interconnection of numerous autonomous systems (AS), each exercising its own administrative policies to reflect the commercial agreements behind the interconnection. However, routing in overlay

☆ This work was supported in part by NSF grant CNS-0721559.
☆☆ An earlier analysis of the transit violations was presented in IEEE ICNP 2006 and the exit violations in IEEE GLOBECOM 2007.
* Corresponding author. Tel.: +1 678 467 2654.
E-mail addresses: srinivasan.seetharaman@telekom.de (S. Seetharaman), ammar@cc.gatech.edu (M. Ammar).

[1] Our work pertains to infrastructure or service overlays, rather than peer-to-peer networks.

1389-1286/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2008.09.014

networks often violates these inter-domain policies to its advantage. For instance, Akella et al. [2] show that a large percentage of overlay routes in the Akamai network [1] violate inter-domain policies to obtain improvement in round-trip time. In this paper, we are interested in investigating the *impact of overlay routing on the enforcement of native network routing policies*. More specifically, we are interested in the extent to which overlay paths violate AS *transit policies* and *exit policies*.

Consider, for example, a hypothetical AS-level connectivity graph as shown in Fig. 1. In that figure, nodes A, B and C are overlay nodes trying to obtain the best possible route to each other. Node B can route data to node C using the overlay path BAC, which results in University X's AS being used for transiting traffic between University Y and Commercial organization Z. This is a violation of the AS transit policy at University X. From an economic perspective, we see that University Y saves money paid to Provider 2, by not using the legitimate route between nodes B and C. This saving comes at the expense of University X, which is not part of any transit agreement. Furthermore, the overlay node in University Y sends commercial traffic on an academic link, which is a violation of the end-user agreement, thereby representing an exit policy violation at University Y. Because overlays operate at the application layer, both these violations typically go undetected by the native layer.

We start our investigation by evaluating a case study overlay network constructed over the PlanetLab testbed [31] which provides insights into the frequency and characteristics of the different violations (results are reported in Section 4). In our dataset, it is interesting to note that close to 70% of the multi-hop overlay routes violate the native layer transit policies and over 87% of the multi-hop overlay routes violate the exit policies. It is worthwhile to observe that these native policy violations are not a serious issue if the overlay traffic is a minor component of the overall traffic. But, in some cases it is already a significant portion of Internet traffic. It is also likely that overlay traffic will experience significant growth in the future.

As awareness of the impact of overlay applications increases, there is an impending drive to incorporate extra complexity at the native layer to manage the overlay traffic [11,28,20]. There have been two types of commercial solutions proposed for both enterprise networks and service provider networks:

- Those that help manage overlay traffic without impacting the user experience [20,6,34].

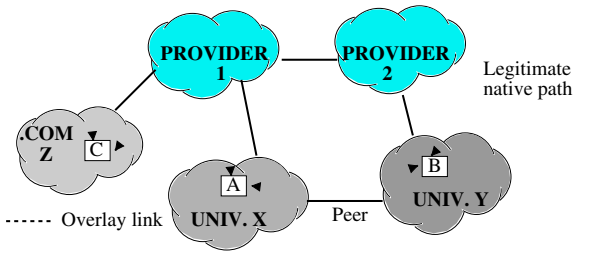


Fig. 1. Policy violations using overlay routing.

- Those that help filter out overlay traffic without concern for the user experience [45,30,40].

This second class of solutions motivates us to investigate the impact of their deployment to counter the inter-domain policy violations committed by overlay routes. Specifically, we focus on two types of overlay traffic filtering – *blind filtering* and *policy-aware filtering*. We show that such filtering can be detrimental to the performance of overlay routing.

There exists two forms of overlays that can cause native policy violations – (i) service overlays (e.g., SON [9], VINI [5], Akamai [1]) where an overlay service provider (OSP) purchases resources from the underlying native layer ISPs in order to offer a value-added network service to actual end-systems and (ii) end-system overlays (e.g., Skype [4]). In the presence of filtering, the user experience will suffer in both these overlays. However, because a service overlay is managed by a single operator, it is feasible for the overlay network to regain the full advantage of overlay routing by adopting one of two approaches we propose. In the first approach, overlay nodes are added so that good overlay paths do not represent inter-domain policy violations. This approach attempts to insure that overlay paths conform to native policy. In the second approach, the overlay acquires transit permits or exit permits from certain ASes that allow certain policy violations to occur but only for permitted overlay traffic. This approach attempts to "legitimize" native policy violations through commercial agreements between the OSP and the native network.

We further develop a single *cost-sharing* framework that allows the incorporation of both these approaches into a single strategy. We formulate an optimization problem that aims to determine how the overlay network should allocate a given budget between paying for additional overlay nodes and paying for transit permits to ASes. We develop a heuristic solution to this problem and illustrate its application on our overlay case study. Further, we evaluate its performance under varying network characteristics.

The remainder of this paper is organized as follows. We define and describe the different types of policy violations possible in Section 2 and classify them in Section 3. We characterize the extent of native policy violations using our case study overlay network and present associated results in Section 4. Section 5 investigates the effect of native layer enforcement of routing policy on the performance of the overlay. We present our cost-sharing approach for mitigating the effect of packet filtering in Section 6. Previous research related to our work are briefly described in Section 7. We summarize this paper in Section 8.

2. Description of policy violations

The current Internet is made up of thousands of autonomous systems that coexist, cooperate, and compete for usage of its various resources. Each AS establishes some form of native layer policy to express its willingness to allow or deny traffic from its neighboring ASes. The Border Gateway Protocol (BGP) is the policy-based routing proto-

col that runs between autonomous systems, implements the various policy constraints and helps each AS select the routes to a destination.

The native network policies are primarily motivated by economic costs and performance gains [29]. These policies predominantly reflect the commercial agreements between ASes and are encoded into the router configuration to be enforced at ingress and egress points of the administrative domain. The combination of these individual policies determine the *AS-path* used by the flows in the Internet; the AS-path is defined as the ordered list of ASes a packet needs to traverse to reach the intended destination. The interconnection of over 20,000 different ASes in the Internet leads to a complex structure with path inflation [44] and unpredictable routing behavior at times [26].

2.1. Definition of policy violations

We define policy violation as the act of using a route at the overlay layer that actually may be objectionable to the ASes involved if used at the native layer. The only reason overlay layer is able to use such a route is because of the misdirection created by overlay relaying, which hides the actual destination from the native layer.

2.2. Types of policy violations

There are different types of native layer policy violations possible, based on what policies the border router enforces. Overlay routing can potentially violate any of these policies at will. We study the violation of two forms of inter-domain policy:

- *Transit policy*: The transit policy (also known as the *valley-free* policy) of AS-paths states that no AS will act as a transit for traffic originating from its provider or its peer, unless the traffic is destined to its customer [13]. We are interested in the violation of the valley-free property because this is the case in which the violation is experienced by an AS not involved in any way with the end-to-end communication. Hence, we consider this to be the most reproachable.
- *Exit policy*: The exit policy is modeled as *the preferred combination of the next hop AS and the egress inter-domain link, for a particular destination IP prefix*. We define exit policy violation as a deviation from this preferred exit caused by overlay relaying. We posit that such deviation causes undesired load and expenses for the native layer, and is objectionable to the native ASes. However, the level of objection to each exit violation may vary with each AS and each overlay path.

Fig. 2 gives an example of both forms of native layer policy violation we noticed during our overlay route measurement process. In this example, the overlay path shown in the figure was determined to be optimal and in particular was superior to the native network path from Colorado State University to the University of North Carolina. We observe that Harvard University is having to use its access bandwidth to/from the Internet to act as a transit between Colorado State University and the University of North Carolina. Thus, the situation might be undesirable for the Harvard University native network, while being beneficial to the overlay nodes at Colorado State University (by providing it an alternate path with 6.3 ms lower latency, a 10.48% gain over the native route). We also observe that Internet2 is forced to use different inter-domain links for sending traffic destined to University of North Carolina. This may be objectionable if it was not previously compensated for that usage.

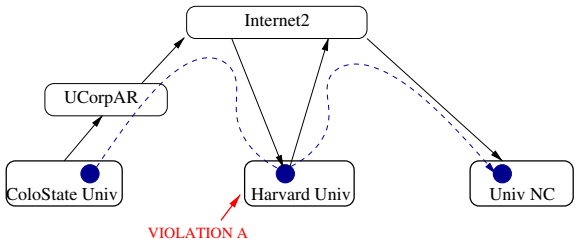


Fig. 2. Example of a plausible violation in reality. The solid circles represent the overlay nodes and the dashed line represents the end-to-end overlay path.

Note that if the overlay node at Harvard University were also a *consumer* of the data being forwarded (in addition to being a relay), we do not consider this a violation since this becomes true application-layer forwarding. In the above example, if the overlay node at Harvard University were a consumer of the data, then it will be part of the communication even in the absence of relaying. More examples of this include end-system multicast, email forwarding and P2P file-sharing, where the intermediate node also uses the content. Hence, a transit violation is when the actual end-to-end AS-path used by the overlay is a violation of the native routing policy and the relay nodes on the overlay path are non-consumers of the data being forwarded.[2]

2.3. Economic model

In this paper, we assume that each overlay node is placed in an AS i by paying a new node fee of N_i ($N_i \geqslant 0$). However, each added overlay node is bound by certain end-user agreements with the hosting AS, which clearly specifies the allowed access. We adopt such a model because the hosting AS in turn incurs expenses (monetary and load) to provide connectivity to this overlay node.

An example of this agreement is one where Internet2 specifies that its customers do not use their network for commercial traffic. As mentioned earlier, overlay routing is quite capable of violating this agreement (or policy). Another example of a violation is when an overlay node within an AS starts providing Internet connectivity to other home users by running a PPP connection over the phone line and surreptitiously forwarding the associated traffic to the Internet. Such examples reveal that the exact policy violation is when the overlay node uses the native network for more than it agreed for.

[2] Our work is restricted to scenarios that are confirmed to be an instance of overlay routing, although it is complicated to verify if the traffic is relayed.

Consider the example in Fig. 3 where overlay nodes are shown as customers of the native network. Although all overlay nodes form a single administrative domain at the overlay layer, they are part of three different administrative domains at the native layer. This causes objections to anarchical behavior from the customer nodes and might cause the hosting AS to revisit the end-user agreement.

In Fig. 3, assume node B connects to $Client_2$'s network by paying a fee of $\$C_B$. This can be for (1) unlimited usage, (2) certain amount of bytes expected from B or (3) certain access bandwidth consumed by B. Note that $Client_2$ in turn pays $Provider_1$ and $Provider_2$ for its Internet connectivity. Hence, the contract with node B will be supplemented by a end-user agreement that states acceptable behavior because $Client_2$ wishes to keep its cost incurred low. In such a scenario, the overlay node B acting as a transit between node A in $Client_1$ and node C in $Client_3$ may cause unnecessary expenses for $Client_2$, for certain end-user contracts. This is true in the case of academic ASes where the overlay node has unlimited usage rights. This explains why violating transit policy at the overlay layer can be objectionable. Furthermore, $Client_1$ may possibly have to use a more expensive ISP $Provider_1$ for traffic that is actually destined to node C. This may be undesirable from the perspective of $Client_1$. This explains why violating exit policy can be objectionable.

The level of objection may be different with each violation. It is possible that for certain end-user agreements there may be no objections from the native layer. In the rest of the paper, however, we assume the worst case scenario, where all violations are serious and are of equal importance to the ASes concerned.

3. Classification of policy violations

In this section, we present our classification of the various forms of transit and exit violations, and highlight some important observations about their nature. Further, we describe the relation that exists between two classes of violations.

3.1. Classification of transit violations

Previous studies on BGP misconfiguration highlighted four different types of route export violations prevalent in the current Internet [26], each of which violates the valley-free property of AS-paths. In the same spirit, we investigate the type of violations caused by overlay routing, which control the route for a certain end-to-end connection by using intermediate relays.

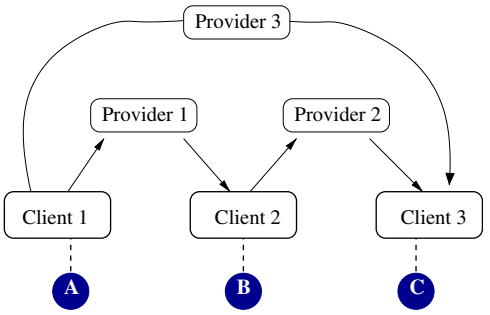


Fig. 3. Illustration of connectivity between overlay network and native network. The solid circles represent the overlay nodes and the dashed line represents the end-user agreements.

Fig. 4 illustrates eight different forms of relaying possible. We argue that cases A, B, C and D represent a violation of native layer policies as the overlay traffic uses the intermediate overlay node to transit through $Client_2$. Thus, $Client_2$ acted as a transit between its provider and peer, which is a violation of the commercial relationships between the ASes. However, cases E, F, G and H do not represent violations because one of the overlay nodes was located in a provider (non-stub) network. In general terms, no violation exists if the native routing policy condones or allows $Client_2$ to be a transit between $Client_1$ and $Client_3$.

3.2. Classification of exit violations

Each AS can exercise different exit policies, two extremes of this policy being hot-potato and cold-potato routing [42]. Fig. 5 illustrates the four basic forms of exit policy violations possible. We describe each as follows:

E1. *Next hop AS violated*: This is caused when the overlay traffic is relayed through an intermediate node located in an AS not along the direct native route between the end nodes. Fig. 5(E1) illustrates a simple scenario where the source overlay node causes $Client_1$ to pick $Provider_2$ to indirectly reach a destination in $Client_2$.

E2. *Ingress or egress router preference violated*: An exit point violation can happen when an overlay path uses a relay node in a downstream AS that is closer to a different ingress router not used by the direct native route. Clearly, this could be a violation of the Localpref attribute, hot-potato routing or cold-potato routing. In Fig. 5(E2), we can see that the local egress preference (router R1) and the neighbor's ingress preference (router R3) are violated, without the knowledge of either AS. This has the problem that the load from the overlay traffic is borne by the link R2–R4 instead of the designated inter-domain link R1–R3.

E3. *Exit point violated because of a next hop AS violation*: This form is similar to the previous scenario, except that the router preference is affected by the alteration of the next hop AS at a downstream provider. In Fig. 5(E3), we see that a change in next hop AS at $Provider_1$ causes a change in the ingress point from R3 to R4. Such a change in ingress router is achieved when $Provider_1$ offers a different MED value for its perceived destination prefix.

E4. *Next hop AS violated because of an exit point violation*: When a downstream AS spans a wide geographical region, it is possible that a change in the ingress point into its domain might cause it to alter its preference of the next hop AS. In Fig. 5(E4), we see that a change in the ingress point from R3 to R4 causes $Provider_1$ to transit traffic through $Provider_2$, rather than sending directly to $Client_2$.

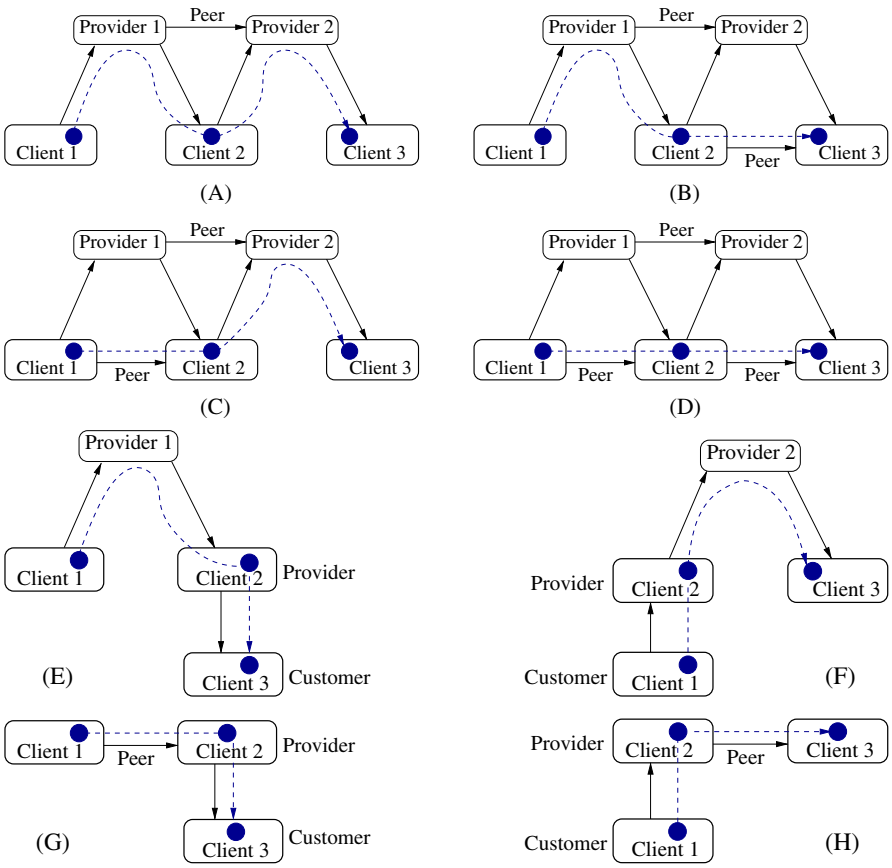


Fig. 4. AS relationships within each overlay path. The solid circles represent the overlay nodes and the dashed line represents the end-to-end overlay path.

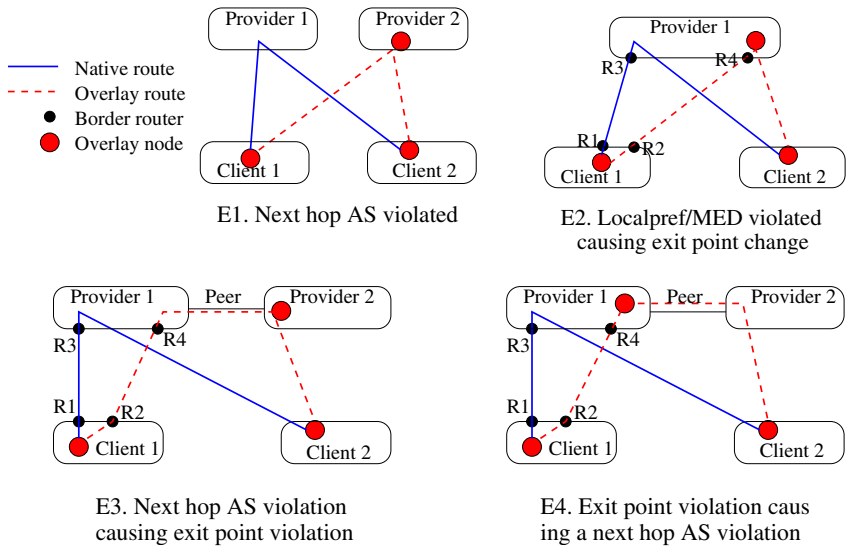


Fig. 5. Possible exit policy violations. Each of these violations can occur at any point of the multi-hop path, either at a host AS or a non-host AS.

The above four scenarios capture the different types of exit policy violations. Each violation has serious economic or load repercussions and is undesirable from the perspective of the native service provider.

The exit violations in each path originate at a single AS where the inter-domain link changes. However, based on the particular type, it may be experienced by other downstream ASes as well. Unlike valley-free violations, the exit

violations are not restricted to only an intermediate host AS.[3]

3.3. Preliminary observations

There are multiple reasons for using a multi-hop overlay path, rather than the direct overlay link (essentially the native network path between two nodes) e.g., achieve better performance or resilience, circumvent limitations imposed by firewalls and NAT boxes, or load balancing, to name just a few. Note that the basic native route between two overlay nodes (referred to as the *overlay link*) never constitutes a policy violation; under the assumption that the native routing conforms to policy. Thus, we make the following observation and investigate only multi-hop overlay paths in the rest of the paper:

Observation 1. *A single-hop overlay path between two overlay nodes does not represent a native transit policy violation.*

An interesting artifact of shortest path routing performed at the overlay layer is that *it is sufficient to analyze each 2-hop overlay path*. Consider an overlay network with nodes A, B, C and D. If the shortest path between nodes A and D is ABCD, then the shortest path between nodes A and C is ABC. Hence, if the 2-hop overlay path ABC or BCD is violating, then the 3-hop overlay path ABCD will also be violating. Furthermore, the number of violations in a multi-hop path is a summation of the violations observed in its constituent 2-hop overlay paths. This confirms that the 2-hop overlay path scenarios considered in Figs. 4 and 5 represent all types of policy violations possible.

Based on the discussion about which scenarios represents a transit violation and which do not, we make the following observation:

Observation 2. *When the overlay nodes are located in stub domains (domains with no clients), every multi-hop overlay path in which the relay nodes are not data consumers represents a transit policy violation.*

From the classification of exit violations, we make the following observation:

Observation 3. *An exit violation can originate at any one of the three following locations:*

- *Source AS (which is also a host AS).*
- *Intermediate host AS.*
- *Intermediate non-host AS.*

3.4. Relation between the two classes of violations

We argue that any overlay path having a valley-free violation necessarily has at least one exit violation at an upstream AS. This can be reasoned by the fact that a valley-free violation occurs at an intermediate host AS that lies outside of the direct native route and such a deviation must originate at an upstream AS (See Fig. 1, for example). Thus, exit violating paths represent the superset of all violating paths. Furthermore, in a particular multi-hop overlay path, the same AS will never experience both exit violations and valley-free violations. This is because valley-free violations happen only at ASes that are not along the legitimate route between the end points, in contrast to exit violations.

We address the following two important questions in the next two sections:

- What is the extent of native layer policy violation in a typical overlay routing situation? – Section 4.
- If the native routing policies are enforced and we disallow certain routes, how much is the overlay routing efficiency affected? – Section 5.

4. Characterizing overlay violations

In this section, we provide insights into the extent of native policy violations in overlay networks. We do this using an experimental case study overlay network deployed over PlanetLab [31].[4] We first describe the overlay case study and investigate the characteristics of its overlay paths. Next, we evaluate the performance gains that overlay routing provides. Lastly, we examine the extent of policy violations that show up in the case study. The overall measurement methodology is summarized in Fig. 6.

4.1. Overlay network case study

We choose 58 PlanetLab nodes that are geographically distributed (based on latitude/longitude) over North America, with only one node per AS. We refer to the AS in which an overlay node is located as the *host AS*. We list the PlanetLab nodes used and the associated native route measurements in [35].

Note that 75% of these overlay nodes are located in educational institutions. We do not believe this to bias the study because the inferences and the solutions proposed in Section 6 apply to any scenario where we observe a significant *number of multi-hop overlay paths*. This number gives an idea of the level of misdirection caused by overlay routing, and tends to be proportional to the level of cross-layer conflict [38]. We anticipate the presence of multi-hop overlay paths in any overlay network that offers significant routing advantage over the native network. The rest of the paper applies to all such overlay networks.

We assume complete mesh connectivity of overlay links between the overlay nodes, for all the results presented in this paper. Following standard terminology, an *overlay link* represents the direct native route between two overlay nodes, which in turn comprises one or more native links, and an *overlay path* is made up of one or more overlay (virtual) links. This overlay path represents the end-to-end

[3] We refer to an AS in which an overlay node is located as the *host* AS.

[4] We present a simulation-based study in Section 6.5 to improve generality of our experiments.

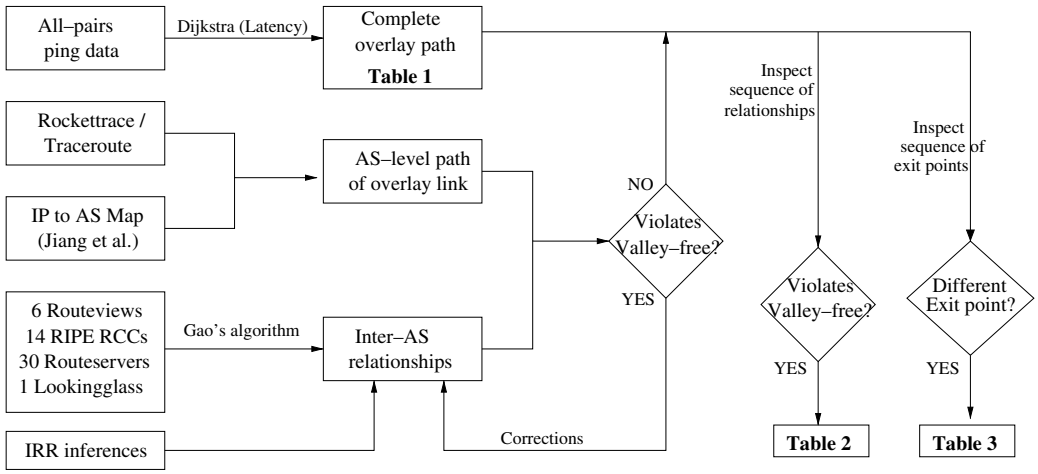


Fig. 6. Measurement methodology used. We use several BGP and PlanetLab measurements to estimate the number of policy violations present in our case study network. Further details provided in Section 4.3.

Table 1
Multi-hop paths in our PlanetLab measurements

(a) Summary of number of multi-hop paths

Metric	Measurement scheme	# Multi-hop paths (of 3306 total paths)	%
Hop count	Scriptroute [41]	394	11.91
Latency	Ping RTT	1868	56.50

(b) Latency-based multi-hop paths

Direct	Hop count of multi-hop paths						
	2	3	4	5	6	7	8
1438	1053	375	315	85	20	17	3
43.5%	31.9%	11.3%	9.5%	2.6%	0.6%	0.5%	0.1%

route taken by the overlay application traffic. There are a total of 3306 overlay paths possible in our topology.

The best overlay path between two overlay nodes may not always be the direct path. In many cases, it is beneficial to route the overlay connection through other overlay nodes, than adopt the direct route [3,10]. Such a decision is typically made by running a routing algorithm at the overlay layer using application-specific routing performance objectives.

For the case study overlay network, we ran a shortest path routing algorithm using hop counts and latency. Table 1(a) presents the different multi-hop overlay paths observed. Table 1(b) further classifies such multi-hop paths when the routing metric is latency. It is interesting to note that almost 56.5% of the overlay paths use a multi-hop route. This provides us ample data to analyze. Moreover, end-to-end latency is a metric that many applications would like to optimize. Hence, through the rest of the case study analysis, we study the violations observed for the particular routing metric of latency.

4.2. Overlay routing performance

Here we address the performance improvement obtained in our case study through the use of multi-hop overlay routes. We quantify the efficiency of overlay routing by using the *gain* metric. The gain achieved for a path is defined as:

$$\text{Gain for path AB} = \frac{(\text{Overlay link latency})_{AB} - (\text{Overlay path latency})_{AB}}{(\text{Overlay link latency})_{AB}},$$

where the $(\text{Overlay link latency})_{AB}$ is the latency of the direct native route between nodes A and B, and (Overlay path

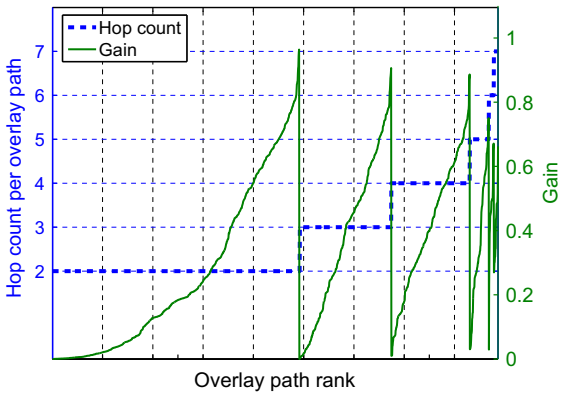


Fig. 7. Distribution of gain corresponding to each hop count value.

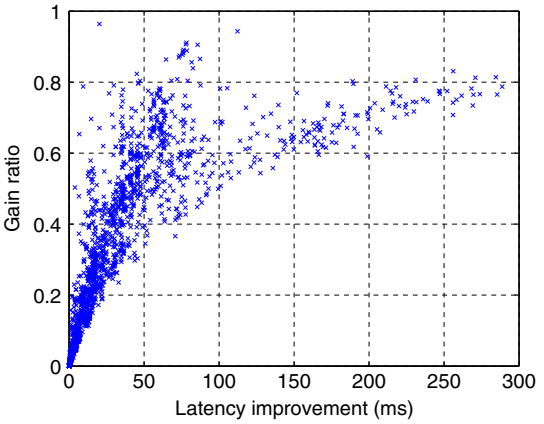


Fig. 8. Scatterplot of latency improvement vs gain ratio in our case study network.

latency)$_{AB}$ is the latency of the shortest path through the overlay network between nodes A and B. Note that (Overlay link latency)$_{AB} \geqslant$ (Overlay path latency)$_{AB}$ always.

The gain metric represents the reduction in end-to-end latency achieved relative to the native route. The value ranges between 0, when the direct overlay link is the optimal one, and 1, when the multi-hop overlay path latency (which is the optimal one) is very small relative to the direct path.

Fig. 7 plots the values of gain observed for each multi-hop overlay path in our data set, sorted based on the hop count of the overlay path and ordered in the increasing order of gain. In the same graph, we plot the corresponding hop count of that overlay path. We observe that the individual curve for each hop count is similar and comparable. Table 2 presents the average and standard deviation of gain observed over paths of a certain hop count. Both the figure and the table indicate that, in our case study, there is not much dependence between the hop count and the gain achieved. In other words, a higher hop count does not indicate a lower gain as one would guess.

Fig. 8 presents a comparison between our gain metric and the absolute latency improvement. We observe two distinct branches in the plot corresponding to a set of low latency paths that have a high improvement in latency, and a set of high latency paths that have a relatively lower improvement in latency. It is conceivable that the absolute latency improvement is also an important metric for some type of applications. However, we use only the relative gain metric through the rest of this paper, as it represents improvement for a vast set of overlay paths and not just a few, i.e. an OSP would be able to procure a bigger revenue if she manages to improve latency for all overlay paths, and not just provide a high improvement for a few paths.

Table 2
Average and Stddev of gain for each hop count value

	Hop count of multi-hop paths					
	2	3	4	5	6	7
Average	0.256	0.38	0.416	0.368	0.458	0.407
Stddev	0.251	0.241	0.204	0.199	0.145	0.109

The *betweenness* of a node is the number of overlay shortest paths that pass through it [12]. Fig. 9 plots the values of (non-zero) betweenness that were observed for each overlay node in our data set, sorted based on the decreasing value of betweenness. We clearly observe a non-uniformity in the relay popularity. From the shape of the betweenness curve, we conclude that there are a few overlay nodes that are the main reason for multi-hop overlay paths being preferred over the direct route. In the figure, we also mark the nodes located in stub domains and those located in non-stub domain. We can note that our dataset has a number of non-stub domain nodes with high betweenness. This is the main reason for the high percentage of non-violating overlay paths (as seen in the following subsection). However, the number of overlay nodes located in non-stub domains in our dataset is not restricted to those indicated in the figure, as the figure plots only nodes with non-zero betweenness that are currently being used in the overlay paths. In Fig. 9, we also plot the value of out-degree for its host AS and its siblings. We do not observe a particular correlation between the betweenness and the AS out-degree, which indicates that overlay rout-

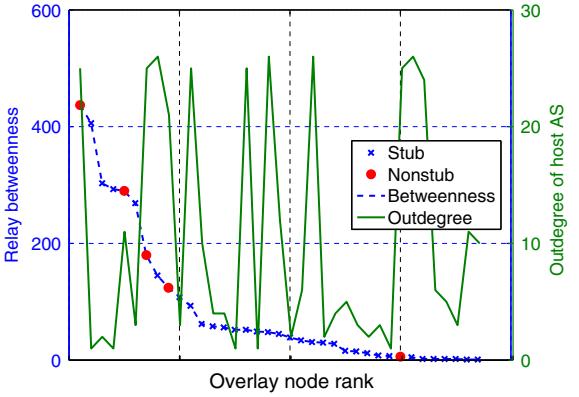


Fig. 9. The betweenness of each overlay node, plotted along with the out-degree of its host AS.

ing is strictly driven by the latency-based edge costs and not the number of AS neighbors.

4.3. Estimation methodology

Estimating the extent of policy violations in our case study requires the end-to-end AS-level path of each overlay path and the individual relationship between each consecutive pair of ASes in the end-to-end AS-path.

4.3.1. Inferring AS path

The Scriptroute [41] data from the PlanetLab measurements provides the IP address of each native hop in the overlay link.[5] We use the publicly available IP-prefix-to-AS mapping generated by the dynamic algorithm in [27]. This work primarily extracts the origin AS of each IP prefix from the BGP routing tables (from sources like the Routeviews servers [33], RIPE servers [32]) and refines the entries further using a dynamic algorithm. By performing a longest prefix match, we obtain the IP-to-AS mapping for each of the native hops. We also cross-verified our IP-to-AS mapping with those generated by *undns* in the Scriptroute tool.

After resolving the AS number of each IP hop in the overlay link, we have the end-to-end AS-path for each overlay path. Any usage of the term AS-path, henceforth, will represent the sequences of ASes in an overlay path, derived by concatenating the AS-path of individual overlay links, unless specified otherwise.

4.3.2. Obtaining AS relationships

In order to obtain AS relationships, we adopt Gao's algorithm [13], supplemented by the partial AS relationship information [46], and our own heuristics to eliminate most of the algorithm's inaccuracies. Gao's algorithm makes inferences based on the AS-paths extracted from the BGP tables and identifies each relationship as being either a customer–provider relationship, a peering relationship, or a sibling relationship. The output from Gao's algorithm is more accurate when we input a more complete view of the AS-paths currently used in the Internet. Hence, as suggested in [47], we obtained the BGP tables from 6 Routeviews servers [33], 14 RIPE RCCs [32], 30 public routeservers and 1 lookingglass server [24].

As the algorithm is forced to use heuristics to classify some of the ASes as a provider, the relationships inferred are not guaranteed to be error-free. Firstly, it has been established that Gao's algorithm does not have a good level of accuracy with inferring peering and sibling relationships [46]. Secondly, the BGP table does not necessarily have information about all the possible inter-AS connections, as some ASes (representing stub networks that are most often simple customers) do not export routes to its peers. To solve these issues, we apply three corrections to the output of Gao's algorithm:

- Partial AS relationship information extracted from the RADB and the RIPE databases of the Internet Routing Registries (IRR) [18]. We followed a procedure similar to that in [46] to obtain these partial relationships.
- Implications from observation 1 that *the AS-path of all overlay links must be valley-free*. For instance, Gao's algorithm inferred that Global Crossing is a customer of Level3 Communications. However, this inference violated the valley-free property of the AS-path of some overlay links. Hence, we resolved the relationship according to what might lead to the valley-free property.
- Hypothesis that unknown relationships between a pair of stub ASes are often unreported peering relationships.

4.4. Transit policy violations observed

We used the AS information obtained above to characterize the transit policy violations in our case study. The statistics are summarized in Table 3(a). Note that each overlay path might commit multiple native policy violations, thereby giving a total of 2629 violations for 1868 multi-hop overlay paths. From the table, we observe that a predominant portion of the violations are of type A (as described in Section 3), followed by those of type B. It is also worth noting that 30.09% of the 3115 intermediate relaying operations performed by the multi-hop overlay paths do not constitute a violation.

We also observed that about 30.19% of the 1868 multi-hop overlay paths do not commit any native policy violation. This is because all intermediate overlay nodes of those paths are located at a non-stub AS. In our dataset, all multi-hop overlay paths with more than 2 relays (3 hops) represent a violation.

Table 3(b) shows the individual percentage of transit violations for overlay paths of different length (in terms of the overlay hop count). It is interesting to note that overlay paths with high hop count display more violations of type A, and rarely any of the other three types, implying that peering relationships are rarely present in long overlay paths in our case study. This could be because the peering relationships do not always offer a path with better latency, but rather offer paths with lower economic cost.

Table 3
Analysis of overlay paths for transit policy violations

(a) Summary of violating relay operations

Type	Description	Number	%
A	Provider–AS–Provider	1925 relays	63.09
B	Provider–AS–Peer	74 relays	2.43
C	Peer–AS–Provider	61 relays	2.00
D	Peer–AS–Peer	73 relays	2.39
None	No violation	918 relays	30.09

(b) Individual % of transit violations for paths of a certain hop count

Type	2	3	4	5	6	7	8
A	41.7	64.7	75.1	76.6	77.6	77.3	80.0
B	5.3	2.7	0.8	0.0	0.0	0.0	0.0
C	3.0	3.4	1.1	0.0	0.0	0.0	0.0
D	5.6	1.0	0.1	3.9	1.0	0.0	0.0
None	44.4	28.2	22.9	19.5	21.4	22.7	20.0

[5] In certain anomalous cases, where the rockettrace did not work at certain nodes, we performed the traceroute operation.

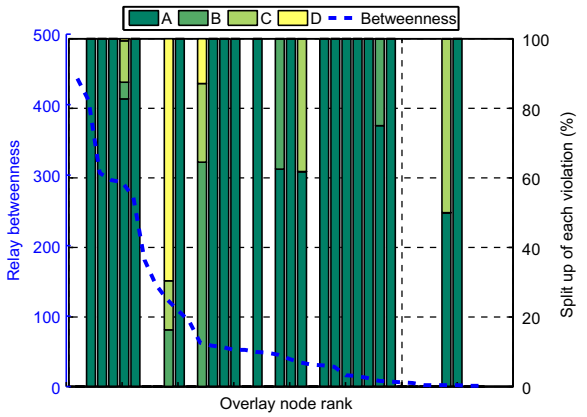


Fig. 10. Partitioning of the 4 different policy violations present at each relay.

Fig. 10 presents the partitioning of the different transit violations observed in the domains that host the intermediate overlay nodes, along with the betweenness of the corresponding overlay node. We observe that most of the overlay nodes with high betweenness have a non-zero number of policy violations and most of these violations are of type A. We can observe from Figs. 9 and 10 that overlay nodes located at non-stub ASes do not commit native policy violations.

Nearly 43% of the transit violating paths are of the form A–B–C–B–D. In this form of violation, the overlay node located at C was only used to route the overlay traffic over AS B, and possibly not for any performance improvement. This violation could, hence, have been avoided during the overlay topology design process. This highlights the importance of overlay node placement.

4.5. Exit policy violations observed

Using the same measurement data, we estimated the frequency of exit violations noticed by comparing the shortest path and the direct native path in the original PlanetLab data. Table 4 presents the summary of exit violations observed. We note that nearly 87.7% of the multi-hop overlay paths represent an exit violation. Interestingly, not all multi-hop overlay paths represent an exit violation as one might expect. This is because the AS path and the exit routers used are the same for almost 12.2% of the overlay paths, though the exact route traversed by the multi-hop overlay path is indeed different from that of the direct native route. This peculiarity is mainly observed when the intermediate node is located in the Internet2 AS. We also affirm from the data that all 1208 multi-hop overlay paths having valley-free violations also have exit violations.

Most of the exit policy violations we observed in our testbed network was of the type where the next hop AS is changed by overlay routing, viz. E1 and E3. Note that for violations of type E3, we consider the change in the next hop AS as the most reproachable and mark the AS experiencing it as the origin point. Hence, we do not see a source AS originating the violation E3. Furthermore, we see more exit violations originating at source ASes (33.2% of total) and at intermediate non-host ASes (51.8% of total) in our dataset.

As mentioned earlier, a violation in a 2-hop overlay path BC is potentially seen in each overlay path AD that overlaps path BC, i.e. the same exit violation might be part of two different paths because the end points are different, although the intermediate segments are the same. Furthermore, a violated source AS in the path BC will count as a violated intermediate host AS for all multi-hop paths AD that overlap this particular path. Note that the results presented above correspond to the total number of violating paths and not number of unique exit violating hops.

Fig. 11 presents the number of exit policy violations experienced by each AS traversed by the multi-hop overlay paths. We observe that the number of exit violations is non-uniformly distributed, such that a small fraction of the ASes is violated by a large number of overlay paths.

Lastly, we analyze the relation between the transit violation and exit violation. When we inspect all exit violations that happen in the overlay link AB because of a transit violation at node B, we notice a stronger correlation between the AS experiencing the most exit violations and the node B, rather than with node A. However, a particular transit violating intermediate relay does not have a unique exit violated AS preceding it, i.e. there is no one-on-one correspondence between the transit violated AS and the exit violated AS. The number of exit violations per AS, associated with a particular intermediate relay, is often distributed in the same non-uniform manner as in Fig. 11, i.e. Some ASes experience significantly more exit violations compared to others. This observation helps us improve the mitigation strategy in Section 6.

Table 4
Exit violations noticed in the PlanetLab dataset, along with the number of paths having valley-free violations

Type	Originating location	Exit violations	%	Valley-free violations
E1	Source AS	502	26.9	444
	Intermediate host AS	104	5.56	76
	Intermediate non-host AS	372	19.9	331
E2	Source AS	43	2.30	1
	Intermediate host AS	113	6.05	0
	Intermediate non-host AS	135	7.22	11
E3	Intermediate host AS	26	5.83	19
	Intermediate non-host AS	342	18.3	326
E4	Source host AS	1	0.05	0
Total violating paths		1638	87.7	1208

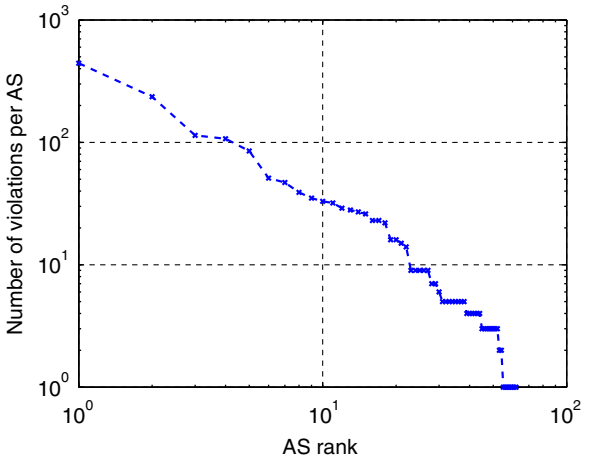


Fig. 11. Log–log plot of the number of exit violations experienced by each of 62 ASes. This shows clear non-uniformity in the violations experienced.

Although this paper only considers the metric of latency at the overlay layer, we expect similar violating behavior with other metrics (e.g., loss rate, throughput) as well, as long as the overlay routing offers substantial improvement over native routing. This similarity can be reasoned by the fact that policy violations are primarily caused by multi-hop overlay paths, which tend to deviate from the direct native route. For instance, Akella et al. [2] show that 17–23% of the overlay paths are multi-hop when the metric of throughput is used to route overlay traffic over the Akamai network. In summary, the higher the number of multi-hop overlay paths (in other words, higher the percentage of relaying), the higher the extent of policy violations.

It is possible that a certain deviation from the standard end-to-end path may not be objectionable to the AS involved. In our dataset, we observed that 61.05% of the deviating overlay paths (43.83% being of type E1) have an AS path that is substantially longer or shorter than that of the direct native route. In such cases, we can assert that the exit violations observed are indeed objectionable. We can say so because an AS that has a choice between two inter-domain routes will first pick based on policy and then based on length. Hence, choosing a path that is substantially different in exit points and AS length is clearly a violation of both decision criteria and definitely objectionable. However, we are unable to establish with confidence the seriousness of the other violations.

5. Effect of filtering on overlay performance

In Section 1, we briefly described the motivation behind various administrative domains (AS) aiming to filter overlay layer traffic. Such filtering is propelled by the negative impact of overlay routing on the native layer, like: defeat of traffic engineering [23,21], route instability [23,21,36], and eventually upsetting the economics of AS interconnection.

This filtering may defeat the purpose of overlay networks and eliminate the flexibility in overlay routing. Nevertheless, it is essential that the native layer have some basic control over such cases of policy violation and then exercise its discretion in determining what can or cannot be allowed. This need is illustrated by the number of commercial solutions that provide such overlay traffic filtering functionality [45,30,40].

We start with the premise that native network filtering is possible (Refer to Section 7 for a survey of existing strategies) and consider two types of filtering – (1) *Blind filtering*, in which an AS blocks all overlay relaying through it, and (2) *Policy-aware filtering*, in which an AS blocks transit overlay traffic only if it violates native routing policies. Note that blind filtering may be easier to implement since it does not require knowledge of native routing policy. In either case, ASes filter only relayed (multi-hop) traffic and do not filter overlay traffic that use the direct native route.

When all ASes perform blind filtering, we observe that no overlay path can take a multi-hop overlay route between two end-points. This makes it essential that all overlay nodes are mesh-connected to ensure that all nodes can reach each other. However, when all ASes perform policy-aware filtering and disallow all types of transit policy violations, we notice from results in Section 4.4 that some (30.19% in our dataset) of the multi-hop overlay paths, which commit absolutely no transit policy violation, are not blocked. Hence, we still maintain the benefits derived from overlay routing. This situation is more desirable for the overlay network and its users. However, if exit violations are also disallowed, then the overlay performance suffers drastically as only few (12.2% in our dataset) of the multi-hop overlay paths are not blocked.

We now evaluate the impact of these two forms of filtering on overlay routing using a *penalty* metric. The penalty incurred for each path is defined as:

$$\text{Penalty for path AB} = \frac{\text{Post} - \text{filtering latency of overlay path AB}}{\text{Best possible latency of overlay path AB, assuming no filtering}}.$$

The penalty value is a good indicator of the negative effect when an intermediate relay node in the shortest overlay path disallows relaying and the overlay traffic is forced to take a longer path.

We next use our PlanetLab overlay, characterized earlier, as a case study to evaluate the effect of filtering on overlay routing performance.

5.1. Blind filtering

In Fig. 12, we plot the average penalty (averaged over all multi-hop overlay paths) that would be incurred if the host AS of a particular overlay node blindly filters out overlay traffic, alongside the value of betweenness associated with

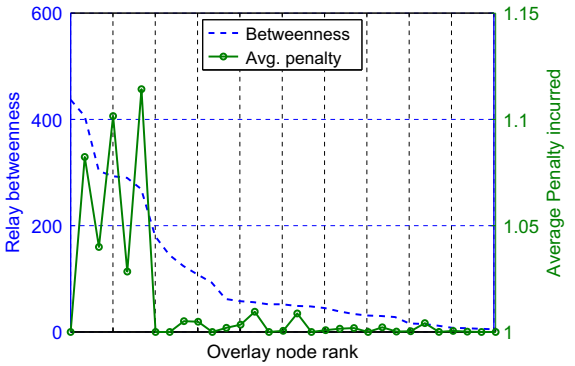


Fig. 12. Performance when overlay relaying is blindly disallowed at a particular host AS, one AS at a time.

that overlay node. To compute the value of the penalty for a particular overlay node, we rerun the shortest path algorithm after disallowing relaying at that node and compute new end-to-end latency values. When an overlay node with high betweenness is disallowed, we seem to incur a high penalty. This indicates that few overlay nodes with high betweenness provide a substantial incentive to each overlay path passing through it and are quite irreplaceable. Hence, depending on whether an excessively used relay is being disabled the overall penalty incurred varies.

Fig. 13 presents plots for the penalty incurred and the number of violations committed when the number of ASes performing this filtering operation is varied. When the number of host ASes performing the filtering operation is n, we have $\binom{58}{n}$ possible scenarios. In cases where this number is over 1000, we chose 1000 scenarios at random. Each penalty measurement in Fig. 13 is an average of the penalty incurred by all overlay paths, and average over all scenarios considered. We also plot the 95% confidence interval for each value.

We observe, from Fig. 13 that, as expected, when more ASes hosting overlay nodes perform blind filtering of overlay traffic, the penalty incurred increases, while the number of violations decreases. We also observe a drastic drop in the number of violations as more than 50 host ASes begin filtering because the last few ASes with high betweenness are finally being targeted.

When all 58 host ASes begin filtering, the overlay paths are forced to use the single-hop overlay link without any relaying, which is equivalent to native routing. This leads to the following three consequences:

- The number of violations observed is 0.
- The penalty value is at a maximum of 1.838.
- There is no advantage to using overlays, as the overlay path is the same as the direct native route.

5.2. Policy-aware filtering

We enforce transit policies on the overlay path by filtering at ASes that host the intermediate overlay nodes of a multi-hop overlay path. We present only the impact of enforcing transit policies because we consider them most reproachable. Furthermore, the impact of enforcing exit policies is similar to that of blind filtering, albeit a bit less detrimental.

Fig. 14 presents plots for the penalty incurred and the number of violations prevalent, along with the 95% confidence interval for each value, when native transit policy violations are disallowed by a certain number of host ASes. We use the same evaluation methodology as that described in the previous subsection. Similar to blind filtering, the penalty incurred increase and the number of violations decrease, with an increase in the number of host ASes performing the policy-based filtering.

When all 58 host ASes begin filtering, the violating overlay paths are forced to use the direct route without any relaying. Hence, the number of violations is 0. However, this does not imply that the gain is 0, as some multi-hop overlay paths are still allowed. In our dataset, we observe an average gain of 13.49% in this scenario (in contrast to 31.81% in the case where there was no filtering). The penalty value is at a maximum of 1.49.

When we compare the results in Figs. 13 and 14, we notice that policy-based filtering incurs substantially lower

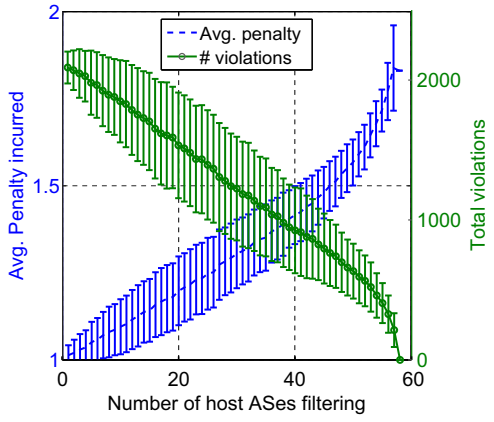


Fig. 13. Blind filtering.

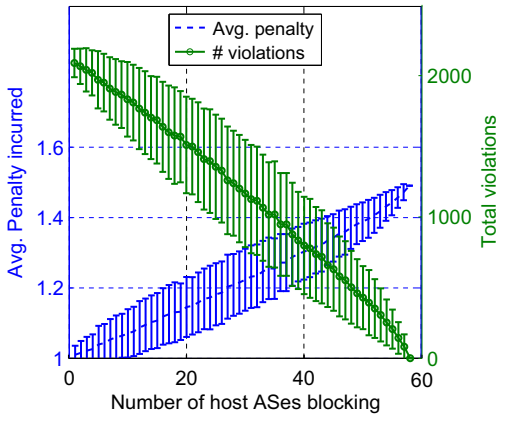


Fig. 14. Policy-aware filtering.

penalty compared to blind-filtering and a non-zero gain, making it still worthwhile to deploy overlays.

6. Mitigating the cross-layer conflict

In Section 4, we investigated the level of inter-domain policy violations. We observed that nearly 70% of the multi-hop overlay paths in our testbed violated the transit policies and nearly 87% of the paths violated the exit policies. This has serious implications on the economics and load experienced by the native layer. In the previous section, we investigated the effect of policy enforcement at the native layer. When the overlay traffic relaying is disallowed at the native layer, we notice a substantial deterioration in performance of the overlay networks. This incapacitation causes the overlay traffic to experience the same or worse treatment as traffic generated by other applications. This indicates a clear *conflict* in objective between the two layers and motivates the need for mitigating this conflict.

We start with a scenario where the overlay network avoids using overlay paths with native policy violations in an effort to appease the underlying native network. This tends to have the same drop in overlay routing gain as filtering does. We next observe that in the context of service overlays that are managed by a common overlay service provider (OSP) it is possible to regain some of the performance advantage, albeit at a cost. From our understanding of the policy violations, we propose two options for improving overlay routing gain:

- Add more overlay nodes at non-stub networks, so that good non-violating overlay paths can be created, as noted from our observations in Section section-classification.
- Negotiate deals with ASes traversed by violating shortest overlay paths (transit and exit), so that overlay traffic is allowed to pass through. This in essence creates an overlay policy that supplements the native policy, but is independently managed. This is the only option available to mitigate exit policy violations because enforcing all exit policies will cause the overlay traffic to take the direct native route.

In this section, we consider a generalized approach where the two schemes above are deployed individually or in combination. In this general scheme, new nodes may be deployed in some parts of the network to create non-violating paths, while ASes in other parts of the network are paid to allow violations (of both transit policy and exit policy). One can also think of the two schemes used simultaneously: an overlay node is added to create some good violating paths and the ASes are paid to allow these paths. By adopting these two approaches, we obtain overlay paths that are better than what is achieved when all ASes perform policy-aware filtering.[6]

This solution allows the OSP to share the cost originally incurred by the native network in return for obtaining a routing performance advantage. Hence, we refer to it as the *cost-sharing* approach. It is conceivable that such an approach can be adopted to relax any objection raised by the native network, thereby fostering a higher level of economic cooperation between the two layers. Adopting this cost-sharing approach is crucial to put an end to the selfish conflict between the two layers, which often leads to a deterioration in routing performance [25].

The basic idea of the cost-sharing strategy is to monetize the objections of the violated ASes i.e., we determine the amount an OSP needs to pay the native layer, so as to use an objectionable inter-domain link or intermediate AS for its traffic. It could be based on the monetary loss incurred by the native service provider for sending traffic in a different direction. Say an OSP pays C_1 to its host AS X for certain usage and causes the host AS to incur an higher expense of C_2 when it performs relaying, then the OSP can pay the overhead of $C_2 - C_1$ to the host AS.

The cost-sharing solution involves the following three costs that are paid by the OSP to the native network, *over the lifetime of the overlay*.[7] Each of these fees can be zero if the overlay nodes are already permitted by the end-user agreement to adopt any overlay paths:

- New node fee, N_i: Cost for adding a new overlay node in native AS i and for the associated network resource usage.
- Transit permit fee, T_i: Cost for making a native AS i allow the transit-violating overlay traffic to be relayed through its network.
- Exit permit fee, E_i: Cost for making a native AS i allow the exit-violating overlay traffic to exit its network.

Note that it is possible the same AS experiences transit violations in one set of multi-hop overlay paths and exit violations in a different set of multi-hop overlay paths. Nevertheless, we devised it such that the OSP pays individual permit fees for better clarity of our approach.

Fig. 15 illustrates the cost-sharing approach in a typical overlay network spread over multiple native ASes. The figure shows the transition from an original network with four violating overlay paths to a network where these paths have been "legalized", by adding a new node to AS_{23} and obtaining a transit permit from AS_{32}. By making these purchases, the overlay service provider obtained 4 multi-hop overlay paths with good performance. Moreover, there were no new violations caused when the new overlay node was added into the overlay topology because AS_{23} is a non-stub domain. Consider for example the route between the overlay node in AS_{34} and that in AS_{33}. The shortest overlay path (AS_{34}–AS_{24}–AS_{35}–AS_{23}–AS_{33}) constitutes a violation of type A and is disallowed by the native layer. This causes the overlay route to adopt the direct native route AS_{34}–AS_{12}–AS_{13}–AS_{33}, which is substantially longer. However, the cost-sharing approach helped obtain a better route through the new node in AS_{23}. This leads to the new overlay route AS_{34}–AS_{24}–AS_{23}–AS_{34} and a corresponding path gain.

[6] We assume that policy-aware filtering will be the dominant solution deployed by ISPs, as OSPs typically purchase a basic service from the hosting ASes to prevent blind filtering.

[7] We avoid usage-based billing to remove effects of traffic variability.

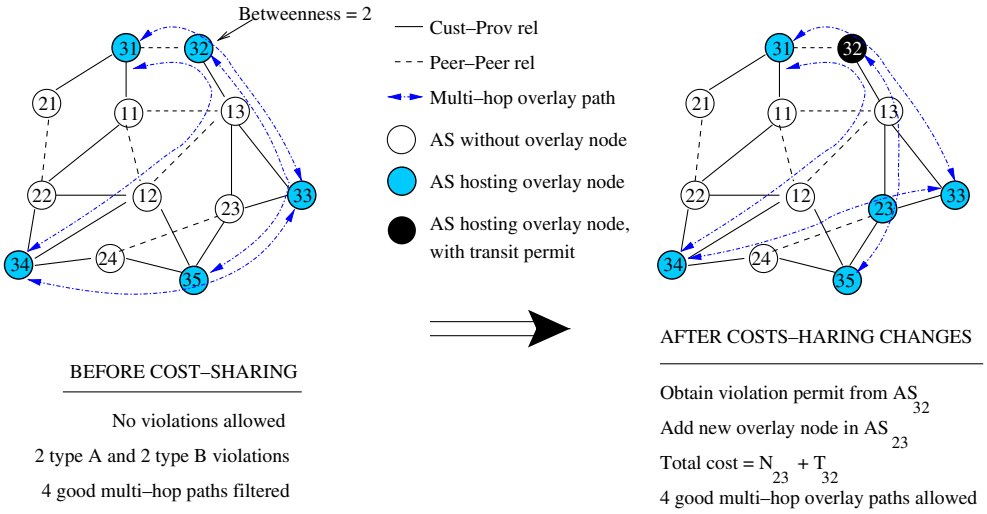


Fig. 15. Illustration of the cost-sharing approach, where we pick the optimal set of ASes to use for relaying. In the above figure, the AS numbers indicate which AS is the provider and which is the customer. For instance, an AS with AS number 12 will be a provider of AS with AS number 22.

However, neither option eliminates the exit policy violation experienced by AS_{34}. The OSP has fewer alternatives for legitimizing the exit policy violations. This is because any deviation from the direct native route, with an objective of attaining a higher routing gain, will cause more exit violations. Hence, the OSP is required to compensate AS_{34} for the exit policy violation in order to retain the routing advantage.

6.1. Deploying the cost sharing scheme

We now consider the question of how to best deploy the cost-sharing scheme described above. The problem we address is the following: Given a certain budget, new node costs and permit costs, how should an overlay service provider determine where to position new nodes and what permits to obtain, in order to maximize its performance advantage within the constraints of the budget.

Based on this problem statement, we can see that the solution to the cost-sharing approach is comprised of two components:

$\mathcal{N}$ = Set of ASes where new nodes are placed.

$\mathcal{T}$ = Set of ASes being paid for transit permits.

$\mathcal{E}$ = Set of ASes being paid for exit permits.

We represent the overall solution as $\mathcal{S}$, where $\mathcal{S} = \{\mathcal{N}, \mathcal{T}, \mathcal{E}\}$. A solution yields a new set of shortest paths in the overlay, $\mathbb{H}$. This set is made up of a set of non-violating or permitted paths, $\mathbb{H}_{\mathbb{N}}$ and a set of violating paths, $\mathbb{H}_{\mathbb{V}}$. The overlay paths in $\mathbb{H}_{\mathbb{N}}$ can provide a gain in routing performance over native routing (as described in Section 4.2). However, the violating paths in $\mathbb{H}_{\mathbb{V}}$ cannot be used (because of the assumption that these are filtered) and the overlay resorts to using the single-hop overlay link. This provides no advantage to overlay routing, i.e. zero gain. An ideal solution, hence, is where $\mathbb{H}_{\mathbb{V}} = \emptyset$.

The cost-sharing deployment problem can be formulated as:

$$\max_{\mathcal{S}} \text{Gain}(\mathcal{S})$$

subject to :

$$\text{Cost}(\mathcal{S}) \leqslant B$$
$$\mathcal{T} \in \mathcal{A}_h$$
$$\mathcal{N} \in \mathcal{A}_p$$
$$\mathcal{E} \in \mathcal{A}_h \cup \mathcal{A}_p,$$

where

B = Budget allocated.

$\mathcal{A}_h$ = Set of host ASes.

$\mathcal{A}_p$ = Set of ASes that are providers to the host ASes, and provider to those providers, and so on.

$$\text{Cost}(\mathcal{S}) = \sum_{i \in \mathcal{N}} N_i + \sum_{j \in \mathcal{T}} T_j + \sum_{k \in \mathcal{E}} E_k,$$

$$\text{Gain}(\mathcal{S}) = \frac{\sum_{i \in \mathbb{H}_{\mathbb{N}}} \text{Gain}(i)}{|\mathbb{H}|}.$$

To solve the cost-sharing problem in the context of inter-domain policy violations and policy-aware filtering, we need the following details about the native network and the original overlay network[8]:

- Overlay network topology (node location, link connectivity).
- Estimated length of each overlay link, based on the metric of choice.
- The AS-level path of each overlay link and the relationships between each pair of AS present in the AS-level path. This helps us determine which relaying operations are filtered by the native layer.
- The hypothetical shortest overlay path computed without concern for inter-domain violations. We denote

[8] Most of these can be obtained by a procedure similar to that we adopt in Section 4.

- Set $\mathcal{N} = \mathcal{T} = \mathcal{E} = \emptyset$
- For each path i in $\mathbb{H}'$
  - For each relay node j in path i, betweenness(j)++
  - Compute Gain(i)
- Sort all overlay nodes in decreasing order of $\frac{\text{betweenness}}{T \text{ for host AS of node}}$
- while total_cost $< B_{th}$
  - j = Get Next Entry(Sorted list of overlay nodes)
  - k = Host AS of node j
  - $\mathcal{T} = \mathcal{T} \cup k$
  - Obtain transit permit from host AS k
  - Compute potential overlay paths $\mathbb{H}$ after obtaining permit
  - In $\mathbb{H}$, determine AS l with highest $\frac{\text{number of exit violations}}{E \text{ for that AS}}$ value
  - $\mathcal{E} = \mathcal{E} \cup l$
  - Obtain exit permit from AS l
  - total_cost = total_cost + $T_k + E_l$
- Sort all overlay paths in $\mathbb{H}'$ with violation A/B in decreasing order of $\frac{\text{path gain}}{N \text{ for upstream provider AS}}$
- while total_cost $< B$
  - i = Get Next Entry(Sorted list of overlay paths)
  - k = Upstream provider AS in path i
  - If AS $k \in \mathcal{N}$, continue
  - If there exists no native route between AS k and destination of path i, continue
  - $\mathcal{N} = \mathcal{N} \cup k$
  - Add new node to provider AS k
  - Compute potential overlay paths $\mathbb{H}$ after obtaining permit
  - In $\mathbb{H}$, determine AS l with highest $\frac{\text{number of exit violations}}{E \text{ for that AS}}$ value
  - $\mathcal{E} = \mathcal{E} \cup l$
  - Obtain exit permit from AS l
  - total_cost = total_cost + $N_k + E_l$
- Solution $\mathcal{S} = \{\mathcal{N}, \mathcal{T}, \mathcal{E}\}$

Fig. 16. Greedy scheme for the cost-sharing problem.

these paths as $\mathbb{H}'$. They represent the highest achievable gain. These are the routes we characterized in Section 4.
- The costs involved in adding nodes and for obtaining permits from ASes, computed from various native-overlay business agreements.

The cost-sharing problem is complicated because of the policy constraints that need to be satisfied. Moreover, the gain value is non-additive with respect to the different ASes used for relaying, i.e., if we know the gain G1 achieved when only AS1 is paid money for relaying and the gain G2 achieved when only AS2 is paid money for relaying, we cannot say that the gain achieved when both AS1 and AS2 are used for relaying is (G1 + G2). This makes our problem different compared to other conventional weight-constrained shortest path problems [16,7,8].

Obtaining an optimal solution $\mathcal{S}$ is a hard problem.[9] Hence, we use insights from our analysis in Sections 4 and 5 to derive greedy heuristics to obtain a reasonable solution.

[9] The cost-sharing problem can easily be shown to be NP-hard by performing a reduction to the set-cover problem, which is known to be NP-complete [14].

6.2. Greedy heuristic solution

Our heuristic solution is shown in Fig. 16. It produces the solution in two phases. In the first, it obtains transit permits for violating paths in a particular order, until the cost of buying permits exceeds a threshold value B_{th}, which is less than or equal to the total budget B. In the second phase, the remaining budget (if any) is used to add new nodes to provide more non-violating paths. The ordering of the two phases is motivated later. During each phase, we simultaneously resolve any exit violations that arise. The order of compensation of the individual ASes in each phase is motivated by the following insights that we obtained from our previous analysis:

1. We observed in Section 4 that most of the violations are in the form of a transit to an upstream provider (Type A and B). Hence, it is desirable to add overlay nodes at these upstream providers (relative to the point of violation), so as to bypass the overlay node associated with the violation. Our heuristic, therefore, adds overlay nodes to intermediate ASes in the unconstrained shortest overlay paths $\mathbb{H}'$, starting

with the violating overlay paths which achieve the highest gain. If there exists an upstream provider in a violating path with very low new node fee N, then it would be in our best interest to give a higher preference to such placing a node there. We achieve this by normalizing the value of path gain by the new node fee for the upstream provider (unless the cost is zero, for which we just use the absolute value), as done in the approximation algorithm for the set-cover problem [17].

2. From the results in Section 4, we know that most of the violations are committed at stub ASes hosting overlay nodes. Moreover, betweenness plots in Section 4 indicated that there are a few overlay nodes that are key to most of the overlay paths. Hence, by merging both observations, we negotiate deals with the stub ASes, in the decreasing order of relay betweenness in $\mathbb{H}'$, to permit the violating overlay traffic to be relayed through. Similar to the previous discussion, we normalize the betweenness value of a particular overlay node by the permit fee for the corresponding host AS.

Adding new nodes or permitting a transit typically has the tendency of changing the adopted overlay path, and thereby the exit points. This indicates that the exit violations must be resolved *after each cost-sharing move*. Our analysis of the location of the exit violations showed no evidence that a particular AS is always violated when a certain overlay node is used in that link. However, we did observe a higher correlation between the location of the exit violation and the intermediate relay used. This further corroborates our choice of estimating exit violations after determining the exact intermediate relay to use at each step.

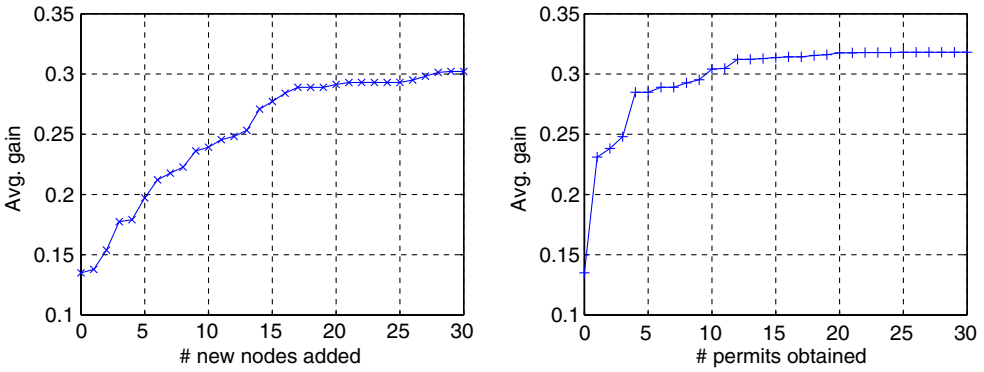
Based on our observations from Fig. 11, we posit that obtaining exit permit from the AS experiencing the most exit violations is the best choice. Nevertheless, if there exists an exit violated AS with very low exit permit fee E, then it would be in our best interest to give a higher preference to obtaining an exit permit from it. We achieve this by normalizing the number of exit violations at each AS by its exit permit fee (unless the cost is zero, for which we just use the absolute value), as done in the approximation algorithm for the set-cover problem [17]. This provides a good tradeoff between budget and path gain.

The exact value we adopt for the budget threshold B_{th} depends on the actual topology of the native and overlay network. When threshold B_{th} = budget B, we only obtain permits to improve overall gain. When threshold $B_{\text{th}} = 0$, we only add new nodes to improve overall gain. This shows that the threshold value B_{th} has a direct influence on the effectiveness of the cost-sharing approach, by controlling the decision of which heuristic to follow. Generally, the ideal threshold value can be found from the betweenness plot in Fig. 9, which shows that only a few nodes are repeatedly present in many overlay paths. Hence, we can look for a knee point in the betweenness curve to determine the appropriate threshold value. Based on the betweenness values observed in our case study and in other simulated overlay networks, we recommend the following rule of thumb for setting the threshold value:

$$B_{\text{th}} \approx \left(\#\text{relays with betweenness} > \frac{\max.\ \text{betweenness}}{2} \right) \times P.$$

6.3. Applying the heuristic to our case study, assuming no exit policy restrictions

In this subsection, we show results from applying our heuristic to the overlay case study analyzed previously. In these results, we assume that the permit fee is the same for all ASes ($T_i = P \forall \text{AS}i$) and the new node fee is the same for all ASes ($N_i = N \forall\ \text{AS}i$). In Figs. 17(a) and (b), we first show the effect of adding nodes or obtaining transit permits in the order specified by the individual heuristic. We make two observations about the heuristics. First, as expected, the ordering of new nodes to add and transit permits to obtain give the desired effect of producing the best gain in the first few nodes added or transit permits obtained. Second, we observe that the gain from the initial few transit permits can be substantial. Hence our decision to obtain transit permits first and then add nodes in the greedy algorithm of Fig. 16.



(a) Only add new nodes (b) Only obtain transit permits

Fig. 17. Average gain achieved with the two individual heuristics.

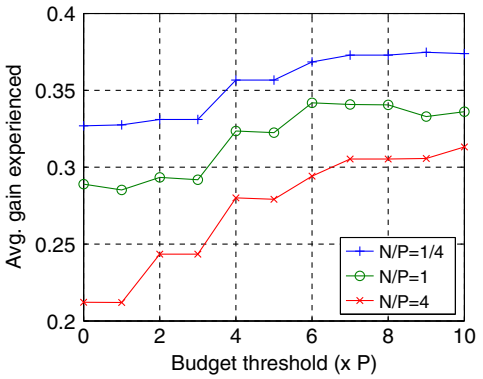


Fig. 18. Solutions for different values of B_{th}, when budget $B = 20 \times P$.

We next applied the cost-sharing algorithm for different values of the threshold value, while keeping the overall budget B at a constant value of $20 \times P$. We plot, in Fig. 18, the solutions obtained for different ratios of N/P. We can see that the knee of each curve lies around a threshold value of 5 or 6. This is coherent with our rule of thumb. We observed in Fig. 17(b) that the gain achieved by obtaining permits saturates after a certain point. Hence, having a high threshold value and filling up the budget with permit expenses is not desirable because we lose on any potential gain that can be achieved by adding new nodes. Keeping this in mind, we varied the threshold value only between zero and $B/2$.

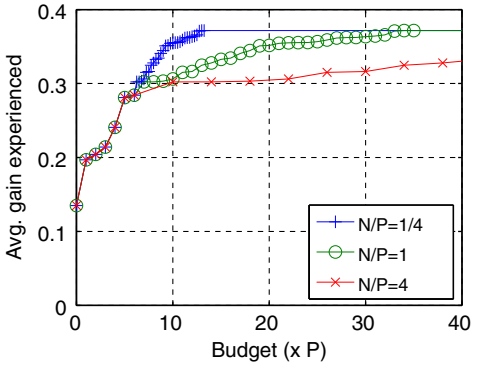
We next consider the effect of the total budget B on the achievable performance. Fig. 19(a) shows the performance when we apply our greedy algorithm for $B_{th} = 6 \times P$, with varying budgets. The ratio between the new node fee and the permit fee determines how effectively the remaining budget will be utilized after obtaining sufficient permits. Therefore, it has a direct bearing on the achieved gain. As expected, for a fixed budget threshold, the higher the N/P ratio, the lower the gain achieved. By comparing the plots in Figs. 17 and 19(a), we observe two important points - (i) the greedy algorithm performs better than the individual heuristics for each value of the budget, (ii) the highest gain achieved in the cost-sharing scheme is greater than what is achieved in $\mathbb{H}'$ (which is the case where all ASes permit violations). These two observations corroborate our combined cost-sharing approach.

All previous experiments assumed equal N_i and T_i. For the same overlay case study, we computed the solution $\mathscr{S}$ for a random distribution of the costs N_i and T_i. The new node fee was uniformly distributed between $[0.5 \times N,\ 1.5 \times N]$ and the permit fees between $[0.5 \times P,\ 1.5 \times P]$, thus maintaining the average values at N and P. Fig. 19(b) presents the gain achieved in this scenario. We observe that the gain achieved for a certain budget is comparable with the earlier simplified scenario. This shows that the algorithm is more influenced by the average costs, rather than the absolute value.

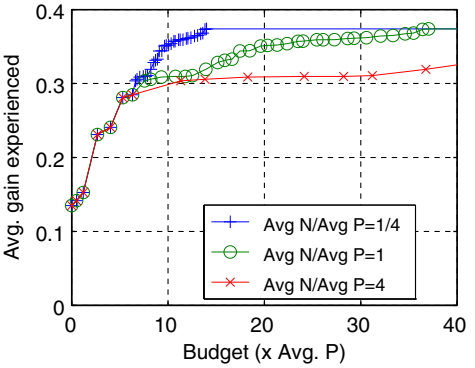
The greedy algorithm in Fig. 16 can easily be amended if the OSP is more interested in the absolute latency improvement of a few overlay paths. Further, by appropriately tuning the threshold, the OSP can determine how much budget to allocate to each part of the cost-sharing solution. We show in Fig. 20 the average of the absolute latency improvement achieved for different budget threshold values, when the P and N for all ASes are equal to a constant c. We observe that adding only new nodes is sufficient for low budgets, while it is best to obtain transit permits when the budget is higher.

6.4. Applying the heuristic to our case study, with exit policy restrictions

We now address the exit policy restrictions. As seen from our greedy algorithm in Fig. 16, the violations are to be resolved simultaneously. For better understanding of the results, we assume the transit permit fee T_i and the exit permit fee E_i for a particular AS i to be equal to P, and the new node fee N_i to be equal to N. Fig. 21(a) presents the gain observed for each expenditure by the OSP, when the budget threshold was configured at $12 \times P$. From the figure, we observe that the OSP is able to achieve a significant improvement in routing performance that is commensurate with the budget spent. We also see that the increase in gain is sluggish when the budget is low. This



(a) When P is equal $\forall$ ASes and N is equal $\forall$ ASes.



(b) When P and N for each AS is uniformly distributed.

Fig. 19. Solutions with the cost-sharing greedy scheme, when $B_{th} = 6 \times P$.

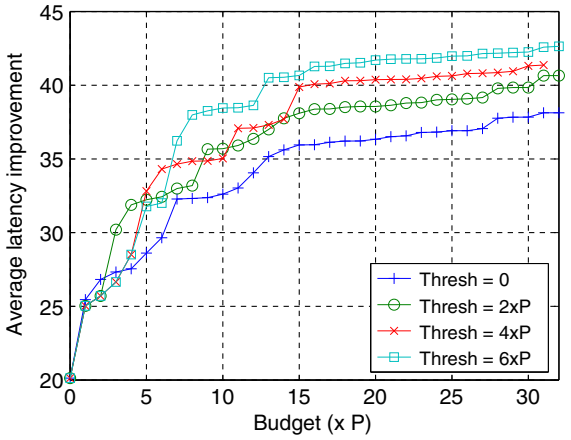


Fig. 20. Absolute latency improvement observed for different values of B_{th}, when $P_i = N_i = c \forall$ ASes i.

is because all ASes in the system filter out violating traffic initially. Obtaining transit permits in that scenario does not cause much change in the gain, until a few critical exit violated ASes are compensated. After crossing that point, the achieved gain increases rapidly with the increase in budget.

Further, we conducted the cost-sharing experiment for a random distribution of the costs N_i, T_i and E_i. The new node fee was uniformly distributed between $[0.5 \times N, 1.5 \times N]$ and the permit fees between $[0.5 \times P, 1.5 \times P]$, thus maintaining the average values at N and P. In this scenario, the improvement achieved for a certain budget was similar to the earlier simplified scenario in Fig. 21(b), showing that the algorithm is more influenced by average costs here as well.

Though there exists no unique exit violated AS that needs to be appeased after each cost-sharing step, our greedy sequential approach offers a reasonable improvement in routing performance for the OSP. Moreover, we do not notice a case where a cost-sharing step fails to make a difference because the corresponding exit violated AS is not compensated.

It is conceivable that this cost-sharing procedure be incorporated from the initial stages of the overlay topology design, rather than serving to supplement an existing overlay topology. This design approach is a hard problem with two unconstrained objectives: (1) Achieving good overall path gain, 2) Resolving arising policy violations. The problem poses different set of challenges as the $\mathbb{H}'$ (set of shortest overlay path without concern for policy violations) does not exist initially and grows with the size of the overlay topology. We reserve further investigation of this problem for future study.

6.5. Network characteristics

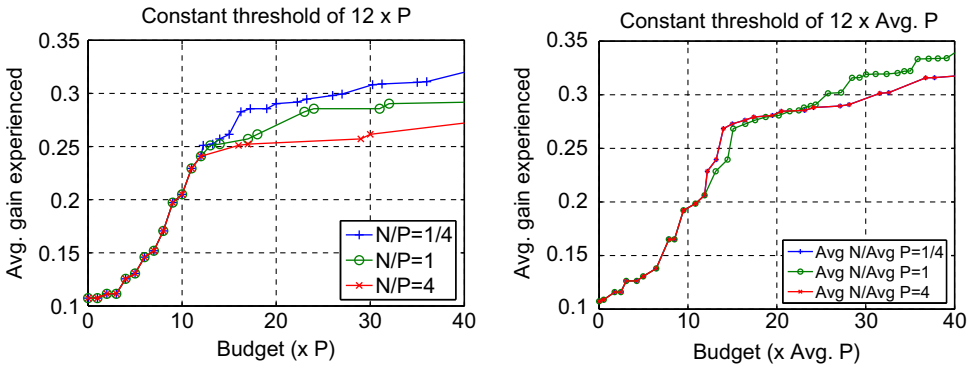
Our cost-sharing approach improves on a given scenario, without creating any new policy violations, by exploiting the following three properties:

1. The property that there exists non-stub ASes that can offer a good route to a destination.
2. The betweenness property of overlay nodes, wherein there exists a small set of overlay nodes that are present in many overlay paths.
3. Exit violations are distributed in a non-uniform manner across multiple ASes.

We understand that many of the conclusions drawn in this section may seem limited by the fact that they originate from a single PlanetLab dataset. In this subsection, we establish the generality of our approach by showing that these three properties hold true in a wide variety of networks.

The first property can easily be reasoned to be true based on the knowledge that inter-domain routing is policy-constrained and does not always adopt the shortest route to a destination. By adding an overlay node at the upstream provider, we are able to force the AS to adopt the shorter route, thereby regaining the routing advantage.

To verify the second and third property, we simulated 90 random overlay networks with varying number of overlay nodes in stub ASes and varying out-degree (extent of multihoming) of the host AS. In particular, the number of



(a) When T and E is equal ∀ ASes, and N is equal ∀ ASes.

(b) When T, E and N for each AS is uniformly distributed.

Fig. 21. Solutions with the cost-sharing greedy scheme with both transit and exit policy restrictions, when $B_{th} = 12 \times P$.

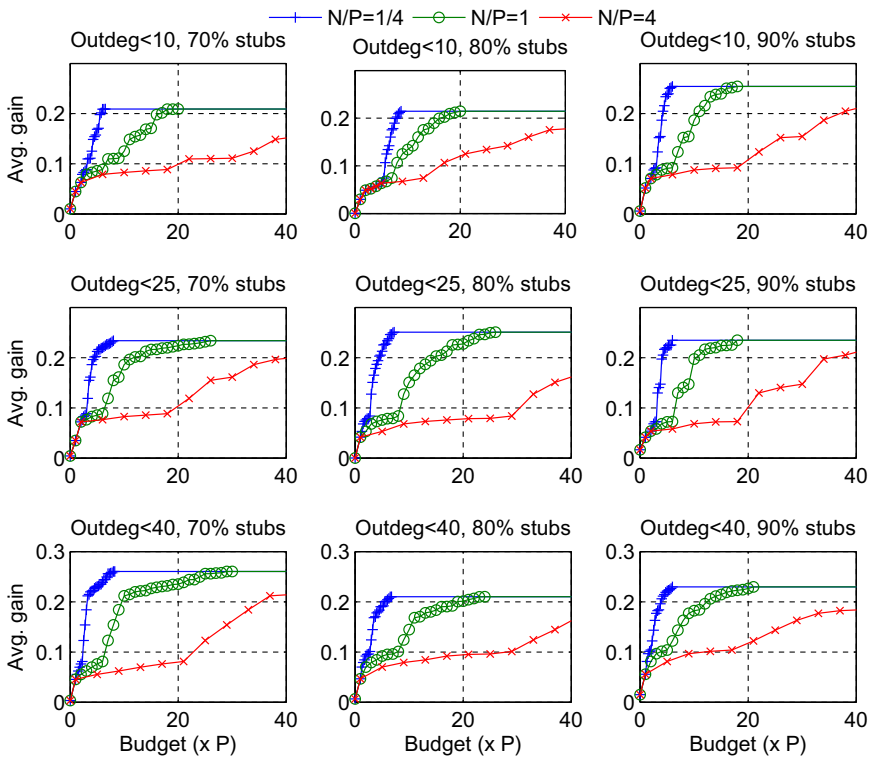


Fig. 22. Cost-sharing solutions for the 9 simulated scenarios.

stub ASes was set at 35, 40 or 45, and the maximum outdegree of the host AS was bound to 10, 25 or 40. This gives a total of 9 combinations, each of which was simulated 10 times.

In each run, we picked 50 host ASes, catering to the above mentioned characteristics, from a list of 21,416 ASes observed in the different BGP route dumps used in Section 4. We computed the AS-level route of each overlay link by using the BGP routes collected from multiple vantage points as input and performing policy routing between the two host ASes.[10] In addition, we assigned random latency values for each inter-AS link (in the range of 10–50 ms) and computed the shortest overlay path between each pair of host AS. When we inspected the betweenness of each overlay node, we observed that the betweenness property indeed holds in all scenarios.

Further, we applied our cost-sharing algorithm to each of the 9 scenarios and plotted the results in Fig. 22. We assumed no exit policy restrictions for this experiment. The gain achieved was averaged over the 10 different overlay networks generated for each scenario. In all scenarios, we notice a sharp increase in the gain when the budget is low. After some point, though, adding more budget does not lead to significant gain improvement. As the initial permits obtained provided a higher gain than what the new node addition provided, our choice of B_{th} (according to the rule of thumb) in each run is justified. The individual plots in Fig. 22 are similar to those observed in our case study, indicating that our approach can be used to improve performance of many possible overlay topologies.

[10] We computed the shortest AS-path that does not violate native layer policy. This is an approximation as the actual routing tables and the policies are unavailable.

7. Related work

Our work is most related to the work by Akella et al. [2], which illustrates the presence of policy violations in real world overlay routes, specifically in overlay routes that improve end-to-end round trip times (RTT) in Akamai's network [1]. They show that 70% of the multi-hop overlay paths violate the transit policy, and an additional 13.3% paths violate the exit policy. This is very similar to what we observe in our PlanetLab testbed, thereby indicating that our results are not an artifact of the testbed. Our work builds over their work to further classify the different types of violations, discuss consequences of native layer policy enforcement, and propose ways to mitigate cross-layer conflict.

Our work is also influenced by, or related to, research in the following four broad directions:

- *Cross-layer interaction in overlay networks*: There are various research efforts that investigate the impact of conflict in objective between the two routing layers, viz. overlay and native. For instance, [25,23,38] investigate the interaction between overlay routing and intra-domain traffic engineering deployed at the native layer.

Their general conclusion is that the interaction causes sustained route oscillations and sub-optimal performance for both layers. Our work investigates the conflict in the inter-domain scenario between selfish latency-based routing at the overlay layer and policies defined at the native layer. Our earlier investigation of the interaction between BitTorrent file-sharing protocol and inter-domain TE shows yet another type of cross-layer conflict [37]. However, the conflict manifests more as a contention for bandwidth between the BitTorrent peers and inter-domain TE.

- *BGP measurement studies*: Our work classifies violations according to results in [26], which tries to understand the BGP misconfigurations that are prevalent in the current Internet. With advances in BGP measurement studies and data sources [13,27,33,32,46,47,24,18,46], it is now possible to determine the different AS policies endured by a packet exchanged between two end systems. We combine both these directions of work in the context of overlay routing to analyze violations of overlay traffic.
- *Native layer traffic management*: The second motivation of our work lies in the rapid development in the market for traffic management products [6,34,45,30,40], based on ASes' need to control the influx of overlay traffic. To identify and filter these overlay packets, most products adopt a flow-signature-based approach [43] that develops some form of correlation between the incoming and outgoing packets, or a communication-pattern-based approach [19]. However, there is very little understanding of the impact of filtering on the user experience. We address this issue in our paper. Karagiannis et al. [20] addresses the impact of P2P networks on ISPs' costs. We address the impact of service overlays on inter-domain policies.
- *Overlay topology design studies*: Numerous studies have investigated the improvement in overlay routing performance achieved by careful placement of overlay nodes and links [39,22]. Our work builds on top of the past research by using the basic overlay topology as an input to our analysis. The work in [39] performs a gain-cost analysis similar to ours, with the aim of picking the least number of servers and achieving the required gain. However, their work does not consider native policy restrictions. Two other efforts on overlay topology design focus on specific routing objectives – Han et al. [15] propose ways to aid the robustness of the overlay network and Zhang et al. [48] propose ways to obtain optimal routing cost and utilization.

8. Concluding remarks

In this paper we investigate the concern that overlay routing derives performance advantages by violating native routing policies. Specifically, we investigated violations of the transit policy and exit policy, caused by multi-hop overlay paths. As more overlay applications are introduced to subvert the functionality limitations of the Internet, the frequency of policy violations can become substantial, which would increase the relevance of this work. Further, we analyze the impact of native layer traffic filtering attempting to prevent these violations. We showed that a clear tradeoff exists between the number of policy violations and the penalty incurred when the native layer enforces these policies. It is conceivable that more networks will start filtering overlay traffic. We showed that even policy-aware filtering can be detrimental to the overlay routing efficiency, while blind filtering can completely remove any incentive to use overlay routing. In this context, we propose a cost-sharing approach that allows the overlay service provider to recover the overlay routing advantage through payments to native network operators. Further, we prescribed a heuristic-based algorithm for solving the cost-sharing problem with a certain budget. We believe that this approach provides a framework to legitimizes native policy violations and allow the benefits obtained by the overlay to be directly related to costs incurred by the overlay service provider.

References

[1] Akamai Technologies, Inc., http://www.akamai.com.
[2] A. Akella, J. Pang, B. Maggs, S. Seshan, A. Shaikh, A comparison of overlay routing and multihoming route control, in: Proceedings of ACM SIGCOMM, 2004.
[3] D. Andersen, H. Balakrishnan, M.F. Kaashoek, R. Morris, Resilient overlay networks, in: Proceedings of 18th ACM SOSP, 2001.
[4] S.A. Baset, H. Schulzrinne, An analysis of the Skype peer-to-peer Internet Telephony Protocol, Tech. Rep. CUCS-039-04, Columbia University, New York, September 2004.
[5] A. Bavier, N. Feamster, M. Huang, L. Peterson, J. Rexford, In VINI veritas: realistic and controlled network experimentation, in: Proceedings of ACM SIGCOMM, 2006.
[6] Cachelogic's VelociX Network, http://www.cachelogic.com/p2p/p2pchoices.php.
[7] J. Clímaco, E. Martins, A bicriterion shortest path algorithm, European Journal of Operational Research 11 (1982) 399–404.
[8] H. Corley, I. Moon, Shortest paths in networks with vector weights, Journal of Optimization Theory and Application 46 (1) (1985) 79–86.
[9] Z. Duany, Z. Zhangy, Y. Houz, Service overlay networks: SLAs, QoS and bandwidth provisioning, in: Proceedings of ICNP, 2002.
[10] S. Savage et al., Detour: a case for informed Internet routing and transport, Tech. Rep. TR-98-10-05, University of Washington, Seattle, 1998.
[11] A. Falk, Not quite the differentiated services I was thinking of, Note sent to e2e mailing list (October 2005).
[12] L. Freeman, S. Borgatti, D. White, Centrality in valued graphs: a measure of betweeness based on network flow, Social Networks 13 (141) (1991) 141–154.
[13] L. Gao, On inferring autonomous system relationships in the internet, IEEE/ACM Transactions on Networking 9 (6) (2001) 733–745.
[14] M.R. Garey, D.S. Johnson, Computer and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman, 1979.
[15] J. Han, D. Watson, F. Jahanian, Topology aware overlay networks, in: Proceedings of IEEE INFOCOM, 2005.
[16] P. Hansen, Bicriterion path problems, in: Multiple Criteria Decision Making: Theory and Applications, LNEMS, vol. 177, Springer-Verlag, Berlin, 1980, pp. 109–127.
[17] D. Hochbaum, Approximation Algorihtms for NP-Hard Problems, Brooks/Cole Publishing Co., 1996.
[18] Internet Routing Registry, http://www.irr.net/docs/list.html.
[19] T. Karagiannis, K. Papagiannaki, M. Faloutsos, BLINC: multilevel traffic classification in the dark, in: Proceedings of ACM SIGCOMM, 2005.
[20] T. Karagiannis, P. Rodriguez, K. Papagiannaki, Should Internet service providers fear peer-assisted content distribution? in: Proceedings of ACM Internet Measurement Conference, 2005.
[21] R. Keralapura, N. Taft, C.N. Chuah, G. Iannaccone, Can ISPs take the heat from Overlay Networks?, in: Proceedings of ACM HotNets-III, 2004.
[22] Z. Li, P. Mohapatra, The impact of topology on overlay routing service, in: Proceedings of IEEE INFOCOM, 2004.
[23] Y. Liu, H. Zhang, W. Gong, D. Towsley, On the interaction between overlay routing and traffic engineering, in: Proceedings of IEEE INFOCOM, 2005.

[24] Looking Glass Servers, http://www.traceroute.org.
[25] L. Qiu, R.Y. Yang, Y. Zhang, S. Shenker, On selfish routing in Internet-like environments, in: Proceedings of ACM SIGCOMM, 2003.
[26] R. Mahajan, D. Wetherall, T. Anderson, Understanding BGP misconfiguration, in: Proceedings of ACM SIGCOMM, 2002.
[27] Z. Mao, D. Johnson, J. Rexford, J. Wang, R. Katz, Scalable and accurate identification of AS-level forwarding paths, in: Proceedings of IEEE INFOCOM, 2004.
[28] E. Mier, D. Mier, A. Mosco, Assessing Skype's network impact, Network World http://www.networkworld.com/reviews/2005/121205-skype-test.html.
[29] W.B. Norton, A business case for ISP peering, White Paper, http://www.equinix.com (February 2002).
[30] Packeteer's PacketShaper, http://www.packeteer.com/prod-sol/solutions/p2p.cfm.
[31] Planetlab, http://www.planet-lab.org.
[32] RIPE Routing Information Services, http://ripe.net/ris.
[33] Route Views Project, http://www.routeviews.org/.
[34] Sandvine's PPE 8200, http://www.sandvine.com/products/p2p_element.asp.
[35] PlanetLab Case Study Data, http://www.cc.gatech.edu/srini/code.
[36] S. Seetharaman, M. Ammar, On the interaction between dynamic routing in the overlay and native layers, in: Proceedings of IEEE INFOCOM, 2006.
[37] S. Seetharaman, M. Ammar, Managing inter-domain traffic in the presence of BitTorrent file-sharing, in: Proceedings of ACM SIGMETRICS (Poster paper), 2008.
[38] S. Seetharaman, V. Hilt, M. Hofmann, M. Ammar, Preemptive strategies to improve routing performance of native and overlay layers, in: Proceedings of IEEE INFOCOM, 2007.
[39] S. Shi, J. Turner, Placing Servers in Overlay Networks, SPECTS.
[40] SonicWALL's Unified Threat Management, http://www.sonicwall.com/products/utm.html.
[41] N. Spring, D. Wetherall, T. Anderson, Scriptroute: a facility for distributed Internet measurement, in: 4th USENIX Symposium on Internet Technologies and Systems, 2003.
[42] L. Subramanian, V.N. Padmanabhan, R.H. Katz, Geographic properties of internet routing, in: Proceedings of the General Track: 2002 USENIX Annual Technical Conference, 2002.
[43] K. Suh, D. Figueiredo, J.F. Kurose, D. Towsley, Characterizing and detecting relayed traffic: a case study using Skype, in: Proceedings of IEEE INFOCOM, 2006.
[44] H. Tangmunarunkit, R. Govindan, S. Shenker, D. Estrin, The impact of routing policy on Internet paths, in: Proceedings of IEEE INFOCOM, 2001.
[45] Verso Technologies' NetSpective P2PFilter, http://www.verso.com/products/netspective/p2pfilter.asp.
[46] J. Xia, L. Gao, On the evaluation of AS relationship inferences, in: Proceedings of IEEE GLOBECOM, 2004.
[47] B. Zhang, R. Liu, D. Massey, L. Zhang, Collecting the Internet AS-level topology, ACM SIGCOMM Computer Communications Review 35 (1) (2005) 53–61.
[48] H. Zhang, J.F. Kurose, D. Towsley, Can an overlay compensate for a careless underlay? in: Proceedings of IEEE INFOCOM, 2006.

Srinivasan Seetharaman is a Senior Research Scientist with Deutsche Telekom Laboratories, USA. He received his Ph.D. in Computer Science from the Georgia Institute of Technology in 2007 and Masters degree in Computer Science from The Ohio State University in 2001. His current research interests include overlay networks, networking architectures and protocols.

Mostafa Ammar is a Regents' Professor with the College of Computing at Georgia Tech. He has been with Georgia Tech since 1985. He received the S.B. and S.M. degrees from the Massachusetts Institute of Technology in 1978 and 1980, respectively and the Ph.D. in Electrical Engineering from the University of Waterloo, Ontario, Canada in 1985. His research interests are in network architectures, protocols and services. He has contributions in the areas of multicast communication and services, multimedia streaming, content distribution networks, network simulation and most recently in disruption-tolerant networks. He was the co-recipient of the Best Paper Awards at the 7th WWW conference for the paper on the "Interactive Multimedia Jukebox" and the 2002 Parallel and Distributed Simulation (PADS) conference for the paper on "Updateable Network Simulation". He served as the Editor-in-Chief of the IEEE/ACM Transactions on Networking from 1999 to 2003. Prof. Ammar is a Fellow of the IEEE and a Fellow of the ACM.