A Fresh Look at Inter-Domain Route Aggregation

João Luís Sobrinho
Instituto de Telecomunicações
Instituto Superior Técnico
joao.sobrinho@lx.it.pt

Franck Le
IBM T. J. Watson Research
fle@us.ibm.com

***Abstract*—We present three route aggregation strategies to scale the Internet's inter-domain routing system. These strategies result from a keen understanding on how the customer-provider, peer-peer routing policies propagate routes belonging to long prefixes in relation to how they propagate routes belonging to shorter prefixes that cover the long ones. The first strategy, *Coordinated Route Suppression*, requires coordination between the Autonomous Systems (ASs) of the Internet, and we present a protocol to perform such coordination. The second strategy, *No Import Provider Routes*, does not require any coordination between the ASs, but benefits only some of them. The third strategy, *Implicit Long Routes*, does not rely on any coordination between the ASs either and it is the most efficient strategy. However, it presupposes modifications to the way routers build their forwarding tables.**

We evaluate the three route aggregation strategies over a publicly available description of the Internet topology and on synthetically generated Internet-like topologies. The results are very promising, with savings in the amount of state information required to sustain inter-domain close to the optimum possible.

I. Introduction

The Internet routing system is in need to scale as its growth and its operational practices—such as the allocation of provider-independent prefixes, multi-homing, and traffic engineering—create an excessive number of prefixes to be exchanged among, stored by, and accessed at its core routers [1]. If the problem is already imperious with IPv4, it can be further aggravated with IPv6, given the added possibilities for segmentation of its larger space [2].

Route aggregation purports to substitute sets of routes pertaining to long prefixes by single routes pertaining to shorter prefixes that cover the long prefixes. We make use of route aggregation to scale inter-domain routing, whereby routing decisions depend on the customer-provider and peer-peer agreements that Autonomous Systems (ASs) establish between them [3], [4]. We consider a short, parent prefix covering each of a number of long, child prefixes in order to present three inter-domain route aggregation strategies: Coordinated Route Suppression; No Import Provider Routes; and Implicit Long Routes. In all three strategies there is an aggregation node that generates a parent route. The aggregation node can be any node whose elected child routes are all customer routes. The *Coordinated Route Suppression* aggregation strategy builds on the observation that if the forwarding-table entry pertaining to a given child prefix coincides with the forwarding-table entry pertaining to the parent prefix, then the former is not needed to expedite data packets [6]. To transform this observation, which is local to a node, to a network-wide route aggregation strategy requires a minimum of coordination between neighbor nodes. The *No Import Provider Routes* aggregation strategy results from the observation that, under certain mild conditions on the topology of the network, a node with an elected child route learned from providers may delete that route and rely, instead, on the parent route. This route aggregation strategy does not require any coordination between nodes, but does not apply to all of them. The *Implicit Long Routes* aggregation strategy is kindled by the subtle observation that whenever a node exports to a neighbor a parent route, without route aggregation it would also export to that same neighbor all child routes. Thus, child routes are implicit in the exportation of the parent route: their explicit exportation is unnecessary. This strategy does not require coordination among nodes, but presupposes that they are able to construct their forwarding tables from implicit knowledge. All three route aggregation strategies are network-wide, decentralized, and preserve the communication paths imparted by the customer-provider, peer-peer routing policies [5].

We evaluate these three route aggregation strategies over an inferred topology of the Internet made publicly available by the Cooperative Association for Internet Data Analysis (CAIDA) [7] and over Internet-like topologies generated with a model similar to the one presented in [8]. The results show very significant savings in the amount of state information required to sustain inter-domain routing.

The remainder of the paper is organized as follows. Section II reviews important facts about inter-domain routing. The three route aggregation strategies are discussed in Section III. Section IV presents experimental results. Section V debates related work while Section VI concludes the paper and points to future research.

II. Inter-Domain Routing

The AS-level structure of the Internet can be modeled as graph, where each node stands for an AS and each link joins two ASs with direct connectivity between them, regulated either by a customer-provider agreement or a peer-peer agreement [3], [4]. Prefixes are allocated to ASs. An AS holding a prefix generates a route pertaining to that prefix that is subsequently propagated throughout the whole Internet by the Border Gateway Protocol (BGP), in compliance with the customer-provider, peer-peer routing policies [5]. These policies prescribe the following: a customer route (learned

978-1-4673-0775-8/12/$31.00 ©2012 IEEE

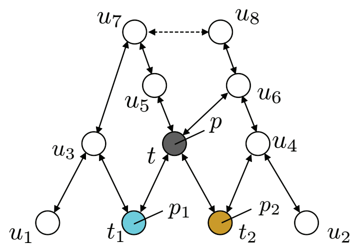

Fig. 1. Solid double arrows join a provider and a customer, with the provider above the customer; dashed double arrows join two peers. Prefix p is a parent prefix with child prefixes p_1 and p_2. Node t_i generates a p_i-route ($i = 1, 2$). Node t is the aggregation node: it generates a p-route.

from a customer) is preferred to a peer route (learned from a peer) which, in turn, is preferred to a provider route (learned from a provider); an AS exports customer routes to all its neighbors, and exports all routes to its customers, these being the only exportations allowed; all exported routes are imported at the receiving AS. We do not consider other attributes of BGP related to routing such as AS-PATH and MED.

Some terminology will prove useful. Routes pertaining to prefix q are called q-routes. Suppose that node z generates a q-route. After propagation by the routing protocol, each node $u \neq z$ either receives a q-route or does not. If u receives a q-route, then its elected (most preferred) q-route can be a customer, a peer, or a provider route. In all three cases, node u installs an entry in its forwarding table, pointing prefix q to the set $f(u; q)$ of its neighbors from which the elected q-route was learned. Such neighbors are called the *q-forwarding-neighbors* of u. If u does not receive any q-route, then its forwarding table does not contain an entry pertaining to q.

Figure 1 depicts a small Internet-like network, where a solid double arrow joins a provider and a customer, with the provider at a higher level than the customer, and a dashed double arrow joins two peers. Node t_1 generates a p_1-route. The elected p_1-route at node u_7 is a customer route. It is learned from both u_3 and u_5 which become the p_1-forwarding-neighbors of u_7, $f(u_7; p_1) = \{u_3, u_5\}$, as shown in each of the sub-figures of Figure 2. (The differences between the three sub-figures relate to different route aggregation strategies to be discussed in the next sections).

We say that a network is *policy-connected* if a q-route generated at any specific node results in an elected q-route at every node. The network of Figure 2 is policy-connected. Hopefully, the Internet is policy-connected as well.

III. Route Aggregation

A. Aggregation Node and Aggregation Coefficient

We focus on a short, *parent* prefix p containing each of a collection of longer, *child* prefixes $p_1, p_2, \ldots, p_N$. The sets of addresses represented by the child prefixes are pairwise disjoint, they are contained in the larger set of addresses represented by the parent prefix, although they need not form a partition of the latter set. The node that generates a p_i-route is denoted by t_i. Given the parent prefix p and its child prefixes p_i ($1 \leq i \leq N$), the goal of route aggregation is to scale the routing processes by judiciously replacing routes and forwarding-table entries pertaining to subsets of child prefixes by single routes and single forwarding-table entries pertaining to the parent prefix, all the while respecting the communication paths that result from the customer-provider, peer-peer routing policies.

An *aggregation node*, denoted by t, is chosen to generate a p-route, which route is propagated throughout the network by the routing protocol, as any other route. The only constraint that we impose on the aggregation node is that its elected p_i-route be a customer route for all $1 \leq i \leq N$. For provider-dependent prefixes, the choice of aggregation node is a natural one. For provider-independent prefixes, the choice of aggregation node may be performed off-line or decided autonomously at a node as long as it satisfies the constraint enunciated above. We leave a full study about the choice of aggregation node, parent prefix, and set of child prefixes for another work.

Regardless of the prefix they pertain to, we say that a customer route is *better* than either a peer or a provider route, and that a peer route is *better* than a provider route. Thus, for example, a customer parent route is better than a provider child route. The cornerstone of our route aggregation strategies, presented in Sections III-B, III-C, and III-D, is the following theorem which establishes a relationship between the elected p_i-routes and the p_i-forwarding-neighbors without route aggregation, on the one hand, and the elected p-routes and p-forwarding-neighbors, on the other.

Theorem 1. *For every node u other than the aggregation node t:*

- *the elected p_i-route is either* better than or as good as *the elected p-route;*
- *if the elected p_i-route is* as good as *the elected p-route, then every p-forwarding-neighbor is a p_i-forwarding-neighbor too, that is, $f(u; p) \subset f(u; p_i)$.*

Our route aggregation strategies are network-wide, scaling forwarding tables, routing tables, and the rate of route exchanges. We measure the efficiency of a route aggregation strategy by the normalized difference between the total number of elected routes without and with route aggregation. Let m_i denote the number of nodes that elect a p_i-route with route aggregation, and m denote the total number of nodes. Assuming the network to be policy-connected, the ratio above is expressed by

$$\frac{Nm - \sum_{i=1}^{N}(1 + m_i)}{Nm},$$

and is called *aggregation coefficient.*

B. Coordinated Route Suppression

The following observation kindles the route aggregation strategy of this section. If, at a node u, the p_i-forwarding-neighbors without route-aggregation coincide with the p-forwarding-neighbors, then u does not need a forwarding-table entry pertaining to child prefix p_i [6]. Without that forwarding-table entry, data packets with address contained in p_i are expedited across the p-forwarding-neighbors, the same neighbors as without route aggregation. However, some other neighbor of u may depend on it to learn a p_i-route and build its own forwarding table. Take Figure 1 as example. The only p_1-forwarding-neighbor of u_5 is t which is also its only p-forwarding-neighbor. Node u_5 does not need to import a p_1-route from t to build its forwarding table. But u_5 is a p_1-forwarding-neighbor of u_7 and the p_1-forwarding-neighbors of u_7, $f(u_7; p_1) = \{u_3, u_5\}$, do not coincide with its p-forwarding-neighbors, $f(u_7; p) = \{u_5\}$. Since u_7 learns a p_1-route from u_5, u_5 needs to import a p_1-route from t after all in order to export it further to u_7.

With a minimum of local coordination, nodes can notify their neighbors when they do not need to learn a p_i-route from them. That coordination is realized with *p_i-suppression messages*, messages that a node sends to another to tell it that it does not need to receive a p_i-route. Node u maintains a variable S_u containing the set of its neighbors that need to learn a p_i-route from it. The nodes of S_u are those to which u has exported a p_i-route and from which it has not received a p_i-suppression message. When u exports a p_i-route to its neighbor x it adds x to S_u. If, later on, u receives a p_i-suppression message from x, then it withdraws x from S_u. When u receives a p_i-route from a neighbor v that does not become a p_i-forwarding-neighbor, it stops importing that route and rather replies to v with a p_i-suppression message, meaning that u does not need to learn a p_i-route from v. Set S_u is empty when no neighbor of u relies on it to learn a p_i-route. If S_u is empty and the p_i-forwarding-neighbors of u coincide with its p-forwarding-neighbors, then u does not need to elect a p_i-route. It stops importing p_i-routes altogether and sends p_i-suppression messages to each of its p_i-forwarding-neighbors.

Let us see the effect of this route aggregation strategy on the network of Figure 1 (see also Sub-figure 2a). We focus on child prefix p_1. Node t_2 has no neighbors to export a p_1-route to. It can reach both t_1 and t via either one of its providers t and u_4, $f(t_2; p_1) = \{t, u_4\} = f(t_2; p)$. Thus, t_2 does not import p_1-routes. It sends p_1-suppression messages to t and u_4. Likewise, u_2 does not import p_1-routes and sends a p_1-suppression message to u_4. Nodes t_2 and u_2 were the only nodes to which node u_4 exported a p_1-route. Since u_4 received p_1-suppression messages from both these nodes, set S_{u_4} becomes empty, and since u_4 reaches both t and t_1 exclusively through u_6, $f(u_4, p_1) = \{u_6\} = f(u_4; p)$, it stops importing the p_1-route learned from u_6 and sends it a p_1-suppression message. Node u_8 exports a p_1-route to both u_7 and u_6, but does not become a p_1-forwarding-neighbor of either of these nodes. They both send it a p_1-suppression message. Therefore, S_{u_8} becomes empty. Because u_8 reaches both t and t_1 exclusively through u_6, $f(u_8; p_1) = \{u_6\} = f(u_8; p)$, it stops importing the p_1-route learned from u_6 and sends it a p_1-suppression message. Node u_6 also receives a p_1-suppression message from t. Having received p_1-suppression messages from all its neighbors, node u_6 sees set S_{u_6} become empty. Because u_6 reaches both t and t_1 exclusively through t, $f(u_6; p_1) = \{t\} = f(u_6; p)$, it will stop importing the p_1-route learned from t and sends a p_1-suppression message to the latter node (which is of no consequence there). The colored nodes of Sub-figure 2a are those that elect a p_1-route.

C. No Import Provider Routes

The following consequence of policy-connectedness triggers the route aggregation strategy of this section.

Theorem 2. *Suppose that the network is policy-connected. If, without route aggregation, the elected p_i-route at a node is a provider route, then any of its providers is both a p_i-forwarding-neighbor and a p-forwarding-neighbor.*

Although the hypothesis of policy-connectedness seems a mild one in practice, it is crucial for the conclusion of Theorem 2 to hold. For example, consider the network of Figure 1 without link u_3u_7. The elected p_1-route at u_1 is a provider route with u_3 being the sole p_1-forwarding-neighbor of u_1. On the other hand, there is no p-route at u_1. Such a route would have to be learned from u_3 and u_3 could only have learned it from t_1. However, t_1 does not export to its provider u_3 the provider route learned from t.

The relevance of Theorem 2 resides in the fact that, under policy-connectedness, the p_i-forwarding-neighbors without route aggregation and the p-forwarding-neighbors are the same at nodes whose elected p_i-routes are provider routes. Therefore, these nodes do not need forwarding-table entries pertaining to p_i. Moreover, if such a node belongs to the set of p_i-forwarding-neighbors of any other node, then the elected p_i-route at the latter node is also a provider route, and can be dispensed with as far as forwarding of data packets with address contained in p_i is concerned. These observations rightly suggest that the paths traversed by data packets with address contained in p_i are unchanged with a route aggregation strategy whereby all nodes whose elected p_i-routes are provider routes stop importing them. We call this route aggregation strategy *No Import Provider Routes*. In Figure 1, nodes u_1, u_2, u_4, and t_2 are exactly the ones that elect a p_1-provider route, as shown in Sub-figure 2b. They can refrain from importing the p_1-routes learned from their providers without distorting the flow of data packets. Especially, node t_2 will still be able to balance data packets with address contained in p_1 between u_4 and t (multi-homing).

The No Import Provider Routes aggregation strategy does not require any kind of coordination among the nodes. On the other hand, it only saves on the forwarding-table sizes of those nodes whose elected p_i-routes are provider routes. In particular, there are no savings to the so called Tier-1 nodes, which are the ones without providers. Being an uncoordinated strategy, it may happen that not all nodes abide to the No Import Provider Routes aggregation strategy at the same time. Yet, nodes have incentives to comply with the strategy. Consider the example of Figure 1 in relation to child prefix p_1 (see also Sub-figure 2b). Node u_4 elects a provider p_1-route, that was learned from u_6. If u_4 does not import that route, then it saves on its forwarding-table

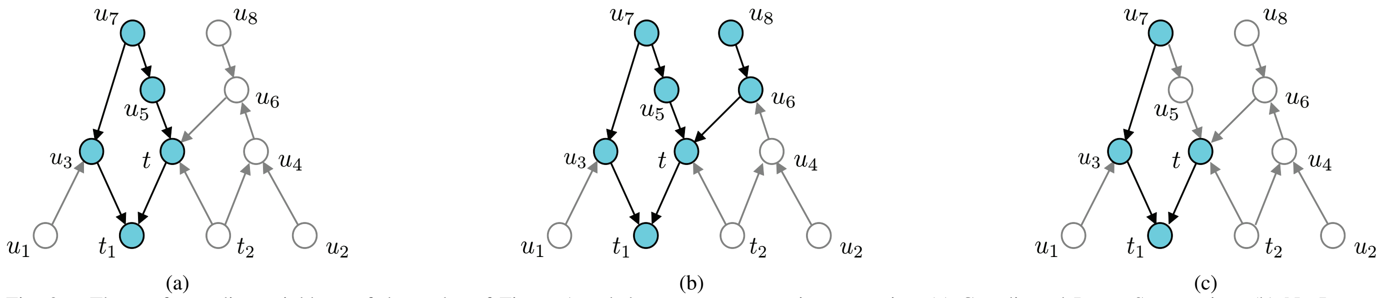

Fig. 2. The p_1-forwarding-neighbors of the nodes of Figure 1 and three route aggregation strategies: (a) Coordinated Route Suppression; (b) No Import Provider Routes; (c) Implicit Long Routes. Colored nodes are those that elect a p_1-route.

size without changing its forwarding of data packets. Thus, u_4 has an incentive to follow No Import Provider Routes. Assume that it does so. Then, u_4 no longer exports a p_1-route to u_2 or to t_2. The forwarding of data packets at u_2 is unperturbed by this absence. That is not so in relation to t_2. This node learns a p-route from u_4 and both a p-route and a p_1-route from t. Because of the longest-match prefix rule, t_2 will start forwarding data packets with address contained in p_1 exclusively to t, thus forsaking multi-homing. However, node t_2 has a double incentive to refrain from importing the p_1-route learned from t. It saves on the size of its own forwarding table and it reverts to balancing the forwarding of data packets with address contained in p_1 across its providers u_4 and t.

D. Implicit Long Routes

Theorem 1 states that the elected p_i-route at any node (other than the aggregation node) without route aggregation is better than or as good as the elected p-route at the same node. In turn, this conclusion implies that whenever a node exports a p-route to a neighbor, without route aggregation it also exports a p_i-route to that same neighbor. This remark suggests that the neighbor can infer the p_i-route from the p-route without explicit exportation of the former, the presence of a p-route standing for itself and for the presence of a p_i-route.

If a node u learns a p-route from every neighbor from which it learns a p_i-route, then u can stop importing p_i-routes. The forwarding-table entry at u pertaining to the parent prefix p points at the same neighbors that would be pointed at by the forwarding-table entry pertaining to the child prefix p_i, since each learned p-route implicitly represents a p_i-route as well. Now, consider a node u that does not learn a p-route from at least one neighbor from which it learns a p_i-route. In this case, node u has to keep a forwarding-table entry pertaining to child prefix p_i and has to build this forwarding-table entry taking into account the meaning of learned p-routes as standing for themselves and for p_i-routes. Specifically, if the elected p_i-route is as good as the elected p-route, then the set of neighbors pointed at by the forwarding-table entry pertaining to p_i must be compounded with the p-forwarding-neighbors. And, if the elected p_i-route is worse than the elected p-route, then the set of neighbors pointed at by the forwarding-table entry pertaining to p_i must be replaced by the set of p-forwarding-neighbors. We call this route aggregation strategy *Implicit Long Routes*.

Consider again Figure 1 with respect to child prefix p_1 and assume that all nodes abide to the Implicit Long Routes aggregation strategy (see also Sub-figure 2c). It is easy to verify that each of the nodes t_2, u_1, u_2, u_4, u_5, u_6, and u_8 learn p-routes from exactly the same neighbors from which they learn p_1-routes. Therefore, each of these nodes can stop importing p_1-routes, and they may perceive an advantage in doing so since it saves on their forwarding-table sizes. Because node u_5 does not elect a p_1-route, it can export none to u_7. Thus, u_7 learns a p_1-route from its customer u_3 and a p-route from its customer u_5. According to Implicit Long Routes, node u_7 builds its forwarding-table entry pertaining to p_1 from u_3 and from its p-forwarding-neighbors, which is just u_5. Ultimately, the forwarding-table entry pertaining to p_1 points at set $\{u_3, u_5\}$, the same set as without route aggregation. Note that u_7 has an incentive to build its forwarding table as described since it allows it to spread data packets with address contained in p_1 over u_3 and u_5. Sub-figure 2c shows which nodes need to elect a p_1-route according to the Implicit Long Routes aggregation strategy.

IV. Results

We present a summary of results related to the performance of the three route aggregation strategies: Coordinated Route Aggregation (CRA); No Import Provider Routes (NIPR); and Implicit Long Routes (ILR). We have realized the strategies both on an inferred AS-level topology of the Internet provided by CAIDA [7] and on synthetic Internet-like topologies generated according a model similar to that presented in [8].

Tier-1 ASs are those without providers. The Tier of any other AS is one plus the Tier of its provider of highest Tier. Stub ASs are those without customers. We randomly (uniform distribution) selected an AS to take the role of aggregation node and randomly (uniform distribution) assigned child prefixes to the stub ASs that can be reached from the aggregation node through a sequence of customer ASs. Table I summarizes the results for four child prefixes. The aggregation coefficients increase with the Tier of the aggregation node, since higher-Tier aggregation nodes correspond to better clustering of the ASs that hold the child prefixes. However, even for Tier-1 aggregation nodes, the aggregation coefficients are very close to the optimum value, which is 0.75 for four child prefixes.

As expected from their description, Implicit Long Routes yields the highest aggregation coefficients, followed by Coor-

TABLE I
ROUTE AGGREGATION COEFFICIENTS FOR FOUR CHILD PREFIXES, AS A FUNCTION OF THE TIER OF THE AGGREGATION NODE.

4 Prefixes		Inferred Internet	Synthetic Topology
Tier 1	CRA	0.718	0.727
	NIPR	0.715	0.725
	ILR	0.719	0.727
Tier 2	CRA	0.731	0.731
	NIPR	0.713	0.727
	ILR	0.732	0.731
Tier 3	CRA	0.742	0.740
	NIPR	0.711	0.730
	ILR	0.743	0.740

TABLE II
ROUTE AGGREGATION COEFFICIENTS FOR 8 AND 16 CHILD PREFIXES WITH TIER-1 AGGREGATION NODES.

Tier 1		Inferred Internet	Synthetic Topology
8 Prefixes	CRA	0.842	0.850
	NIPR	0.840	0.848
	ILR	0.842	0.850
16 Prefixes	CRA	0.911	0.916
	NIPR	0.899	0.914
	ILR	0.912	0.916

dinated Route Aggregation, with No Import Provider Routes remaining for last, although the aggregation coefficients of the first two aggregation strategies are barely the same. The results presented in table I indicate that the size of the forwarding tables, and the number of routes to be stored and exchanged in the Internet can be significantly reduced through route aggregation.

Table II presents aggregation coefficients for when the number of child prefixes increases to 8 and to 16. The results are close to the optima of 0.875 and 0.938, respectively, further confirming that the proposed aggregation strategies can significantly scale the Internet's inter-domain routing system.

V. Related work

References [6], [9] propose *Forwarding Table Aggregation*. This is a technique local to each node which consists in the identification, in its forwarding table, of parent and child prefixes pointing at the same set of neighbors. The forwarding-table entries pertaining to these child prefixes can be deleted without disturbing the flow of data packets. This approach reduces the size of the forwarding tables, on account of their post-processing after the usual updates that arise from routes elected by the routing protocol, but it does not scale the routing processes that sustain the forwarding tables. In contrast to [6], [9], our route aggregation strategies do not require any post-processing of forwarding tables, except, possibly, at the aggregation node. More significantly, our route aggregation strategies, especially No Import Provider Routes and Implicit Long Routes, scale the whole routing processes, not just the forwarding tables.

VI. Conclusions and future work

We proposed novel route aggregation strategies to scale the Internet's inter-domain routing system. Given a parent prefix and set of child prefixes, we choose for aggregation node any one node that elects a customer route to reach each of the child prefixes. Coordinated Route Suppression is based on the observation that if a child prefix's forwarding-neighbors at a node coincide with the parent prefix's forwarding-neighbors, then the forwarding-table entry corresponding to the child prefix is not needed. To construct a route aggregation strategy from this forwarding-plane observation, some coordination is required between nodes. No Import Provider Routes is based on the observation that if the network is policy-connected, then a node that elects a provider route to reach a child prefix ends up not needing that route after all. It is an uncoordinated strategy which nodes have incentives to embrace. The boldest proposal is Implicit Long Routes. It is based on the observation that if a parent route is exported from a node to a neighbor, then a child route would also be exported from the former to the latter node without route aggregation. Thus, child routes are implicit in parent routes, not needing to be exported explicitly. This strategy is uncoordinated, provides the highest performance, but relies on a node's ability to build its forwarding table taking implicit child routes into account. All three route aggregation strategies remain faithful to the communication paths the result from the customer-provider, peer-peer agreements that govern inter-domain routing and all yield aggregation coefficients that are close to the optimum.

A number of issues remain for further inquiry. We highlight two of them here. First, we considered a simple address hierarchy consisting of pairs parent-prefix, set-of-child-prefixes. This hierarchy can be extended to a full address tree with multiple levels of descendants (or ascendants) younger than children (older than parents). Second, we did not address the robustness of the proposed route aggregation strategies to link failures and additions. It turns out that Coordinated Route Suppression requires extra coordination to deal with failures and additions, but that No Import Provider Routes and Implicit Long Routes are inherently robust to failures.

Acknowledgments

The work of João Luís Sobrinho was partly supported by FCT project PEst-OE/EEI/LA0008/2011.

References

[1] V. Fuller and T. Li, "Classless inter-domain routing (CIDR): The internet address assignment and aggregation plan," August 2006, RFC 4632.

[2] D. Meyer, L. Zhang, and K. Fall, "Report from the IAB workshop on routing and addressing," September 2007, RFC 4984.

[3] G. Huston, "Interconnection, peering and settlements - part I," *Internet Protocol Journal*, vol. 2, no. 1, pp. 2–16, March 1999.

[4] ——, "Interconnection, peering and settlements - part II," *Internet Protocol Journal*, vol. 2, no. 2, pp. 2–23, June 1999.

[5] L. Gao and J. Rexford, "Stable Internet routing without global coordination," *IEEE/ACM Transactions on Networking*, vol. 9, no. 6, pp. 681–692, December 2001.

[6] X. Zhao, Y. Liu, L. Wang, and B. Zhang, "On the aggregatability of router forwarding tables," in *Proc. INFOCOM 2010*, March 2010, pp. 848–856.

[7] "The CAIDA AS relationships dataset," January 2010, http://www.caida.org/data/active/as-relationships/.

[8] A. Elmokashfi, A. Kvalbein, and C. Dovrolis, "On the scalability of BGP: the role of topology growth," *IEEE Journal on Selected Areas in Communications*, vol. 28, pp. 1250–1261, October 2010.

[9] V. Khare, D. Jen, X. Zhao, Y. Liu, D. Massey, L. Wang, B. Zhang, and L. Zhang, "Evolution towards global routing scalability," *IEEE Journal on Selected Areas in Communications*, vol. 28, pp. 1363–1375, October 2010.