Seventh FRAMEWORK PROGRAMME
FP7-ICT-2009-5 - ICT-2009-1.6
Future Internet experimentally-driven research

SPECIFIC TARGETED RESEARCH OR INNOVATION PROJECT

Deliverable D2.2
"Routing Scheme and Design Specification"

Project description
Project acronym: **EULER** Project full title: **Experimental UpdateLess Evolutive Routing** Grant Agreement no.: **258307**
Document Properties
Number: FP7-ICT-2009-5-1.6-258307-D2.2 Title: Routing Schemes and Design Specification Responsible: Dimitri Papadimitriou (ALB) Editor(s): Dimitri Papadimitriou (ALB) Contributor(s): D.Papadimitriou, B.Sales, I.Caragiannis, P.Canellopoulos, S.Sahhaf, W.Tavernier, N.Hanusse, C.Glacet, J.-C.Delvenne, D.Careglio, L.Fabrega, P.Vila, M.Camelo, F.Tarissan Dissemination level: Public (PU) **Date of preparation:** *Date* **Version: 0.9**

D2.2 - Routing Schemes and Design Specification

List of authors

Affiliation	Author
ALB	D.Papadimitriou, B.Sales
CTI	I.Caragiannis, P.Canellopoulos
IBBT	S.Sahhaf, W.Tavernier, D.Colle
INRIA	N.Hanusse, C.Glacet
UCL	J.-C.Delvenne
UPC	D.Careglio, L.Fabrega, P.Vila, M.Camelo
UPMC	F.Tarissan

Table of Contents

List of authors 2
Table of Contents 3
1. Introduction 4
2. Internet Routing System and Protocol Architecture 6
2.1 BGP Architectural Limits 6
2.2 Identifying root causes for the absence of suitable alternatives to BGP 7
2.3 Operational Feedback 8
2.4 Bottomline 9
3. Positioning against routing space 10
3.1 Incremental improvements to BGP 11
3.1.1 Stability-based route selection criteria 11
3.2 Path-vector routing 16
3.2.1 Route Verification 16
3.3 New routing paradigms 17
3.3.1 Components 17
3.3.2 Compact Multicast Routing 19
4. Conclusion 23
References 24
Annex 1: Paper "Modeling the Internet Routing System and Protocol Architecture" 25
Annex 2: Paper "Stability metrics and criteria for path-vector routing" . 35
Annex 3: Paper "Relationship between path-vector routing and forwarding path stability" 44
Annex 4: Paper "Design and Performance Analysis of Dynamic Compact Multicast Routing" 51

1. Introduction

Considering that the foundational principles of the Internet routing system are i) distribution (local computation of the routing table entries), ii) adaptivity (to topology and policy dynamics) and iii) policing (decision process, routing updates filtering, etc.), the most fundamental challenges faced nowadays by its architecture are: scalability, adaptation cost, convergence time, and stability of its underlying protocols. These elements are further documented in Annex 1 (paper titled "Modeling the Internet Routing System and Protocol Architecture"). These challenges result from i) the increasing number of routing entries and thus routing states amplified by the design and usage of the addressing system (including prefix de-aggregation practices for traffic engineering purposes, and site multi-homing), ii) the short-term topology and policy dynamics and iii) the longer-term topology evolution (increasing meshedness). Their combination together with the intrinsic limits of the BGP architecture and its underlying properties leads to a very complex problem.

In order to address these challenges altogether, we have modeled in Task 2.1 (and documented in Deliverable D2.1) the routing system by identifying its functional components, their relationships, and their spatio-temporal distribution (functional model) together with the information properties, operations and relationships (information model). The spatial distribution of the above-referenced elements (function and information) is characteristic of distributed systems like the Internet and the temporal distribution of these elements (in particular, the information) is the main source that explains the complexity in specifying an alternative routing protocol to BGP. Note also that in distributed systems like the Internet, the routing decisions are locally performed by each (abstract) node independently of the others using the exchanged information (i.e., discovered information) but individual nodes decision affects other router's decision. It is therefore fundamental to capture these interactions as part of the functional model up to the level appropriate for further routing system and protocol engineering.

This approach enables also to thoroughly identify which routing (sub-)functions are currently under-specified or mis-specified but also which routing (sub-)functions can be replaced, added or even removed from their specification as documented in existing scientific literature. Comparison between the different routing schemes (and associated operations) is also facilitated as functional modeling offers at the same time a detailed functional analysis grid. Complementarily, the information model provides the mean to perform a detailed information analysis. Finally, it is to be emphasized that demonstrating performance gain and improvements and on the other hand proving efficiency as well as utility are equally important.

The present document follows and further develops the architectural approach documented in Task 2.1; it also exploits the results obtained from several activities conducted in the context of WP3. It then details the relationship between the Internet routing system and protocol architecture together with the current limits that can be identified by applying the method developed in Task 2.1 to identify architectural limits of BGP. After identifying the root causes for the absence of suitable alternatives to BGP, the operational feedback obtained from the first Technical Advisory Board (TAB) is analyzed in order to understand the necessary and sufficient conditions for protocol migration. The next section documents and positions the proposed novel incremental improvements to BGP, the research work dedicated to path-vector routing together with the work dedicated to the investigation on genuine alternative to BGP in particular for what concern multicast routing.

2. Internet Routing System and Protocol Architecture

2.1 BGP Architectural Limits

The following table provides an overview of BGP coverage against the functionality outlined here above showing that the base functionality is mostly covered by the Border Gateway Protocol (BGP) besides security. The latter still remains a major operational concern for the operational community.

Table I. BGP functionality coverage

Routing functional area	**BGP Coverage**
Any-to-any connectivity	Yes
Distribution with asynchronous messaging/processing	Yes
Adaptivity - Topology dynamics - Policy dynamics - Traffic dynamics	 Yes Yes Limited
Policy	Yes but node-based decision only
Security	Secure channels, information verification is ongoing (cf. SIDR effort)
Advanced functionality	**BGP Coverage**
Traffic engineering	Limited (mainly by means of spatial/prefix de-aggregation)
Multicast	Yes but requires specific extensions (MP-BGP)

Applying the functional and information analysis to BGP leads to the following crucial observations:

- The routing information exchange process is asymmetric: the RIB_In is actually decoupled from the Loc_RIB whereas the RIB_Out is driven by the selection/update rate of routing entries. The subsequent addition of a threshold to the routing update rate (i.e., the MRAI) at the sender-side is certainly a direct transposition of the "be liberal in what you accept and be conservative in what you send" design principle but in the meantime, the ratio RIB/FIB (function of the number of BGP peering sessions per BGP speaker) can easily reach an order of 10 (if not more since the number of BGP peering sessions is independent of the number of physical interfaces). Thus, routers have often to process an order to 10M routing entries to derive about 450k active routing table entries. Remember that the BGP update process "pushes" routing updates to neighbors. This mechanism defines probably the most basic technique for routing (data)base synchronization but its simplicity may actually be the root cause of the memory size scaling and adaptation cost observed nowadays. This observation leads to possibly rethink the routing update distribution process and not (only) the route selection process.

- The BGP route selection being driven by a node-based decision process, little flexibility is left to update neighbors on a per neighbor-basis beside application of outbound filters. This design is certainly desirable for inbound BGP speakers (with respect to the flow of routing updates) peering with BGP routers belonging to the same AS but less robust for outbound routers peering with different AS's.

- The nature of the routing update information (and its distribution process) is prone to induce path exploration; the question that stems though is why selecting a route subject to path exploration at first place. The answer is essentially because i) routing update information processing does not differentiate between updates with respect to their root cause, their identification (origin), etc. during the route selection process, and ii) the route selection process itself performs solely by applying network-wide criteria on the spatial properties of the AS-Path attributes (carried in routing updates) that are assumed to be immutable when processed. Thus, in addition to the routing update process itself, the information it distributes would have to be extended to incorporate temporal and infer causal properties.

- The BGP route selection process performs "on-path" regarding the flow of routing information updates. This design choice seriously compromises the possibility for introducing any simple routing information verification mechanism crucial for security reasons (as a routing path and its associated routing update flow are congruent). Such mechanism aims at enabling the receiving BGP router to verify that the originating AS is authorized to advertise an address prefix by the holder of that prefix, whether the originating AS is accurately identified by the originating AS Number (ASN) in the advertisement, and the validity of both the address prefix and the ASN.

2.2 Identifying root causes for the absence of suitable alternatives to BGP

One of the main root causes of the absence of suitable alternative to BGP resides in the lack of architectural modeling of the global routing system when designing a routing protocol and its associated routing algorithm(s). Indeed, such design is to be performed in accordance to the routing system and addressing model describing their components and relationships (and not independently). Next, the procedures for routing information exchange and routing path computation can be specified and their impact on the global routing system can be analyzed and evaluated by using the architectural model. Following a systematic architectural method does not specify how to implement the routing procedures and data structures themselves. However, the proper exploitation of this method enables to systematically determine and analyze the composition and the different relations between these procedures and data structures as well as the functional and the behavioral properties these procedures would have to satisfy in order to ensure that the Internet routing system meets its objectives. When the routing system is not properly modeled, the impact of these design choices on the global routing system is almost impossible to evaluate beforehand making any improvement a trial-and-error experiment. Moreover, experience shows that without well-defined routing system architecture, adding/removing or replacing routing functionality increases its architectural complexity.

As explained here above, prominent research efforts have been conducted over last decades to address the challenges related to the Internet routing system. However, their design tends to follow (at least since so far) the exact same approach as the one pursued by BGP. This statement is corroborated by the following observations:

- The routing algorithm still exclusively determines the behavior of the routing system whereas a proper method would assume that the routing system architecture (which comprises a non-local information acquisition function) determines which class of algorithms produces the needed output from the available input (and under which conditions);
- Certain performance objectives are verified by the routing algorithm but without accounting for their dependency on the spatial and temporal properties of the information/input and running conditions (e.g., memory space consumption is minimized in stationary conditions up to the point that adaptivity cost and convergence time objectives become unachievable);
- The functional distinction between the routing information acquisition function being either explicit (push/pull) or implicit (local inference) and the routing path computation function is often neglected. On the other hand, little work has been realized since so far in terms of architectural modeling of the Internet routing system with the purpose of deriving alternative routing schemes (and subsequently routing protocols).

This situation has led to a deadlock in terms of routing research since approaching the problem space requires the design of routing algorithm(s) and protocol but also the specification of the routing architecture that couples both information and functional model. We argue that failing to work simultaneously and in symbiosis with these three dimensions altogether explains for a large part the reason why no suitable alternative to the BGP-based routing system has been proposed since so far but also why only a limited number of the numerous improvements to BGP have been deployed since so far. The other reason is the lack of analytical model translating the behavior (in particular, the spatio-temporal properties) of the entities inducing network dynamics but on which the behavior of the routing system, the routing protocol and the routing algorithm strongly depends.

2.3 Operational Feedback

Following the first Technical Advisory Board (TAB) meeting of the EULER project that was organized on June 8, 2012 at Ghent University, the main operational concerns relate to routing system and protocol functionality. This functionality include in order of priority: adaptivity, policing, and security that BGP offers today (if we would include security considerations as developed in SIDR IETF Working Group). Moreover, incentives for migration to a new routing model/protocol shall be justified by at least one order of magnitude of performance improvement (e.g. memory size) WITHOUT deteriorating other functionality and/or performance currently provided by BGP.

This statement indicates that routing protocol migration would be primarily driven by i) additional functionality; following Table I the extend to which these improvements can be actually considered is rather limited; and ii) as core/edge routers can accommodate O(1M) IPv4 active routes (in Loc_RIB) and O(10M) routes in the Adj_RIB_In, there is sufficient headroom at current deployment rate; in other terms, there should be a significant increase in the routing table growth rate (and associated dynamics) to justify such migration.

Moreover, if the replacing routing model/protocol would induce the use of a different current locator space (compared to the current IPv4/IPv6 space) further justification in terms of reduction of operational complexity and cost (beside the cost of migration) but also new functionality shall be offered as deep operational impact would follow. Such migration would be justified only if the new routing protocol offers new "business opportunities" and not only improves the cost of scale/performance of the Internet routing system. Internet routing protocol is and remains mainly "problem-driven" and not exclusively driven by improvements of protocol performance aspects.

For this purpose, the main interest expressed covers i) the investigation of the IPv6 routing table growth (compared to IPv4) and impact on routing system, ii) the stability of the routing system/routing paths, and iii) the heterogeneity of the environments where BGP can be deployed and perform (assuming extension in data centers for instance).

2.4 Bottomline

Performance improvement is not a sufficient condition for migration to a new routing protocol (assuming that routing protocol would exist) and ***functional preservation if not improvement is a necessary condition.***

Moreover, next to the functional aspects, capturing the spatio-temporal properties of the routing information (as none of the alternative routing algorithm has its pre-conditions verified to provide its output) becomes the main blocking point. This observation is critical in the context of the project because acting at both functional/procedural and information level is required to expect finding an alternative routing protocol. This further corroborates the relevance of the approach followed in Task 2.1 and guides three main directions for designing an alternative routing scheme:

- **Capture as part of the protocol formats, the non-deterministic nature of the routing information, i.e., their variation over time (distribution functions instead of time-invariant numeric values or symbols).**

- **Combine routing information discovery function with the computation procedure(s) but do not inter-twin them; specify a routing information distribution protocol i) by means of communication sessions that are not necessarily congruent with the selected paths (as mandated by the BGP node decision process) in order to enable exchange of information not necessarily used locally by the computation function and ii) that supports information exchange that can operate in hybrid push-pull mode (so as to avoid the drawbacks associated to push-only discovery protocols).**

- **Design the computation function such as to support different rates of arrivals of routing information (multi-modality) while limiting the increase in computation complexity.**

3. Positioning against routing space

Prominent research efforts have been conducted over last decades to address the challenges related to the Internet routing system and its underlying routing protocol. These efforts can be classified as follows: i) incremental improvements to BGP, ii) new class of path-based routing protocols, and iii) new routing paradigms. It is also important to mention that we do not consider as part of the new routing paradigm class known mechanisms which for various technical reasons were never used (e.g., source routing) or never deployed (e.g., hierarchical link-state routing).

In the following, we use the same classification to position our effort and activities aiming at specification of:

- *Incremental improvements to BGP*:
  - Stability-based route selection criteria
  - (Partial) route-verification process

- *Path-vector routing*[1]:
  - The (partial) route verification process being generic, it can expectedly also be applied to new path-vector routing protocols.

- *New routing paradigms*:
  - As stated above since the design of alternative routing scheme requires new foundational elements that still remain to be identified, the possible exploitation of certain components related to routing schemes such as stochastic routing (as it enables processing of routing information subject to uncertainty and adapts accordingly), greedy routing (which builds a set of local routing entries whose memory size is proportional to the degree of each node if we exclude the memory mobilized for storing the results of the operations for coordinate assignment), geometric routing (which operates by assigning to nodes (virtual) coordinates in a metric space; these (virtual) coordinates are then used as addresses to perform point-to-point routing in this space) but also compact routing (for the introduction of the notions of variable neighborhood and coloring or classes of names).
  - Segmentation between the discovery of routing information and the computation/selection algorithm, discovery function that if specified generically enough can also be applied to path-vector routing (but also to any routing scheme requiring non-local knowledge of the topology and its properties to operate). As the focus of the project is on distributed adaptive routing, this new approach for the distribution of routing information is documented as part of the components to be considered in the context of new routing paradigms.
  - Designed independently of the underlying unicast routing protocol, the compact multicast routing scheme that has been developed can run on top of any unicast routing topology. The proposed approach is competitive against both existing compact multicast routing schemes and existing IP multicast routing protocols such as PIM (RFC 4601) or Multicast BGP (RFC 2858).

[1] Note that examples of path-vector routing protocols are documented in the paper reproduced in Annex 1

3.1 Incremental improvements to BGP

3.1.1 Stability-based route selection criteria

Following the work initiated in Task 3.2, we have defined several stability metrics to characterize the local effects of BGP policy- and protocol-induced instabilities on the routing tables (for more details see Annex 2 -

Our experimental results show that the proposed method enables to locally detect instability events that are affecting routing tables' entries, and deriving their impact on the local stability properties of the routing tables. We have also defined a differential stability-based decision criterion that can be taken into account as part of the BGP route selection process.

After documenting needed preliminaries, the following sections document the novelty of the proposed method relying on the definition of a new BGP route selection rule derived from the differential stability metric, the experimental results verifying the Consistency of the stability-based selection criteria, and the selection rule itself and its usage.

i) Prior work and Novelty of the Proposed Method

Numerous studies on BGP dynamics properties have been conducted over last twenty years. Work began in the early 1990s on an enhancement to the BGP called Route Flap Damping (RFD). The purpose of RFD was to prevent or limit sustained route oscillations that could potentially put an undue processing load on BGP. At that time, the predominant cause of route oscillation was assumed to result from BGP sessions going up and down because established on circuits that were themselves persistently going up and down. This would lead to a constant stream of BGP update messages from the affected BGP sessions that could propagate through the entire network. The first version of the RFD algorithm specification appeared in 1993, updates and revisions lead to RFC 2439 in 1998 [1].

Mao et al. [2] published in 2002 a paper that studied how the use of RFD, as specified in RFC 2439, can significantly slowdown the convergence times of relatively stable routing entries. This abnormal behavior arises during route withdrawal from the interaction of RFD with "BGP path exploration" (in which in response to path failures or routing policy changes, some BGP routers may try a sequence of transient alternate paths before selecting a new path or declaring the corresponding destination unreachable). Bush et al. [3] summarized the findings of Mao et al. [2] and presented some observational data to illustrate the phenomena. The overall conclusion of this work was to avoid using RFD so that the overall ability of the network to re-converge after an episode of "BGP path exploration" was not needlessly slowed.

More recently, solutions such as the enhanced path vector routing protocol (EPIC) [4] propose to add a forward edge sequence numbers mechanism to annotate the AS paths with additional “path dependency’’ information. This information is combined with an enhanced path vector algorithm to limit path exploration and to reduce convergence time in case of failure. EPIC shows significant reduction of convergence time and the number of messages in the fail-down scenario (a part of the network is disconnected from the rest of the network) but only a modest improvement in the fail-over scenario (edges failures without isolation). The main drawback of EPIC is the large amount of extra information stored at the nodes and the increase of the size of messages. Another solution, BGP with Root Cause Notification (RCN) [5] proposes to reduce the BGP convergence delay by announcing the root cause of a link failure location. This solution also offers a significant reduction of the convergence time in the fail-down scenario. However, the convergence time improvement achieved with RCN is modest on the Internet topology compared to legacy BGP (in the fail-over scenario). More advanced techniques such as the recently introduced Path Exploration Damping (PED) [6] augments BGP for selectively damping the propagation of path exploration updates. PED selectively delays and suppresses the propagation of BGP updates that either lengthen an existing AS Path or vary an existing AS-Path without shortening its length.

All these approaches try to mitigate instability effects and/or to accelerate convergence after occurrence of a perturbation event, but none of them ask the fundamental question why selecting a route subject to path exploration at first place. The answer is essentially because none of these mechanisms perform rely on the actual quantification of the instability effect and still use network-wide spatial criteria that for AS-Path selection.

ii) Preliminaries

The autonomous system (AS) topology underlying the routing system is described as a graph G = (V,E), where each vertex (or abstract node) $u \in V$, $|V| = n$, represents an AS, and each edge $e \in E$, $|E| = m$, represents a link between an AS pair denoted (u,v), where $u, v \in V$. Each AS comprises a set of physical nodes referred to as routers; the AS representation of the topology combines thus both its partitioning and its abstraction. The subset of physical nodes of interest for this paper comprises the routers running the path-vector algorithm (typically sitting at the periphery of each AS). At each of these routers, a route r per destination d ($d \in D$) is selected and stored as an entry in the local routing table (RT). The total number of entries is denoted by N, i.e., $|RT| = N$. A route r_i to destination d at time t is defined by $r_i(t) = \{d, (v_k=u, v_{k-1},…,v_0=v), A\}$ with $k > 0 \mid \forall j, k \geq j > 0, \{v_j, v_{j-1}\} \in E$ and $i \in [1,N]$, where $(v_k=u, v_{k-1},…,v_0=v)$ represents the AS-Path, v_{k-1} the next hop of v along the AS-Path from the abstract node u to v, and A its attribute set. Let $P_{(u,v),d}$ denote the set of paths from node u to v towards destination d where each path p(u,v) is of the form $\{(v_k=u, v_{k-1},…,v_0=v), A\}$. A routing information update leads to a change of the AS-Path $(v_k, v_{k-1},…,v_0)$ or an element of its attribute set A. Next, a withdrawal is denoted by an empty AS-Path (ε) and $A = \emptyset$: $\{d,\varepsilon,\emptyset\}$. According to the above definition, if there is more than one AS-Path per destination d, they will be considered as multiple distinct routes.

BGP being in the context of this paper the path-vector routing protocol considered; we further detail its storage data structures, referred to as Routing Information Bases (RIBs), used to store its routes $r_i(t)$. At each BGP speaker, the RIB consists of three distinct parts: the Adj-RIB-In, the Loc-RIB, and the Adj-RIB-Out. The Adj-RIB-In contains unprocessed routing information that has been announced to the local BGP speaker by its peers. The Loc-RIB which corresponds to the BGP local routing table (RT) contains the routes that have been selected following the local BGP speaker's decision process. Finally, the Adj-RIB-Out organizes the routes for announcement to specific peers. When a router receives a route announcement, it first applies inbound filtering process (using some import policies) to the received routing information. If accepted, the route is stored in the Adj-RIB-In. The collection of routes received from all neighbors (external and internal) that are stored in the Adj-RIB-In defines the set of candidate routes (for that destination). Subsequently, the BGP router invokes a route selection process - guided by locally defined policies - to select from this set a single best route for each destination. After this selection is performed, the selected best route is stored in the Loc-RIB and is subject to some outbound filtering process and then announced to all the router's neighbors. Importantly, prior to being announced to an external neighbor, but not to an internal neighbor in the same AS, the AS path carried in the announcement is prepended with the ASN of the local AS.

We introduce in [7] the definition of *differential stability* between the most stable route in the Adj_RIB_In and the selected route stored in the Loc_RIB for the same destination d characterizes the stability of the currently selected routes for a given destination d against most stable routes as learned from upstream neighbors. The corresponding metric provides a measure of the stability of the learned routes compared to the stability of the currently selected route. A variant of this metric, denoted $\delta\varphi_i(t)$, $i \in [1,|D|]$ where D is the total number of destination prefixes, characterizes the stability of the newly selected path p*(u,v) at time t for destination d against the stability of the path p(u,v) that is stored as time t in the Loc_RIB for destination d and that would be replaced at time t+1 by the path p*(u,v): $\delta\varphi_i(t) = \varphi_i(t) - \varphi_i*(t)$. In turn, if the differential stability metric $\delta\varphi_i(t) > 0$, then the replacement of route $r_i(t)$ by the route $r_i*(t)$ increases the stability of the route to destination d; otherwise, the safest decision is to keep the currently selected route $r_i(t)$ stored in the Loc_RIB.

iii) Consistency of the stability-based selection

Application of the differential stability metric $\delta\varphi_i$ during the BGP selection process would prevent replacement (in the Loc_RIB) of more stable routes by less stable ones but also enable selection of more stable routes than the currently selected routes. However, for this assumption to hold, we must also prove the consistency of the stability-based selection with the existing preferential-based route selection model that relies on a path ranking function (i.e., a non-negative, integer-value function λ_u, defined over $P_{(u,v),d}$, such that if $p_1(u,v)$ and $p_2(u,v) \in P_{(u,v),d}$ and $\lambda_u(p_1) < \lambda_u(p_2)$ then $p_2(u,v)$ is said to be preferred over $p_1(u,v)$). The route selection problem is consistent with the stability function $\delta\varphi(t)$, if $\forall$ u $\in$ V and $p_1(u,v)$ and $p_2(u,v) \in P_{(u,v),d}$ (1) if $\lambda_u(p_1) < \lambda_u(p_2)$ then $\delta\varphi(t) = \varphi_1(t) - \varphi_2(t) \geq 0$ and (2) $\lambda_u(p_1) = \lambda_u(p_2)$ then $\delta\varphi(t) = 0$. We show in [7] that if $p_1(u,v)$ and $p_2(u,v) \in P_{(u,v),d} \wedge p_2(u,v)$ is embedded in $p_1(u,v)$, then the route selection problem is consistent with the stability function $\delta\varphi$ and the route selection is not stretch increasing. By stretch decreasing, we mean here that the length $\rho_i*(t)$ of the path p*(u,v) (measured in terms of number of AS hops in case of BGP route) associated at time t to the route r_i* is smaller than the length $\rho_i(t)$ of the path p(u,v) associated at time t to the route r_i: $\delta\rho_i(t) = \rho_i*(t) - \rho_i(t) < 0$.

iv) Experimental Verification

The results obtained (see Fig.1) shows that the cumulated percentage of routes with respect to the AS-path length difference between the selected and the most stable route. A positive difference indicates that the replacement of the selected route (using the BGP path ranking function) by the most stable route would decrease the AS-path length compared to the selected route ($\delta\rho < 0$). A negative difference indicates that such replacement would increase the AS-path length ($\delta\rho > 0$).

From this figure, we can deduce that such replacement would be advisable for about 90% of the selected routes since $\delta\rho \leq 0$. Moreover, for 25% percent of the routes, this replacement would also lead to an AS-path length decrease since for these routes $\delta\rho < 0$. Interestingly, only 10% of the routes would be affected by a length increase if they would be selected based on the stability criteria since for these routes $\delta\rho > 0$. Among this percentage of 10%, we can also observe from this figure that a significant fraction of the routes would be covered if an AS-path length increase of one-hop would be considered as acceptable (in average $\delta\rho \cong 1.15$). These observations corroborate the fact that the stability-based selection rule does not lead to a stretch increase for a significant fraction of the routes (90%). On the other hand, by admitting a stretch increase corresponding to one additional AS-hop in the AS-path, only a minor fraction of the routes (about 2%) would be penalized by a higher stretch increase of two AS-hops (and above for a fraction of routes << 1%). This observation can be seen as the experimental evidence that enforcing stability would not come at the detriment of increasing the stretch of the AS-paths.

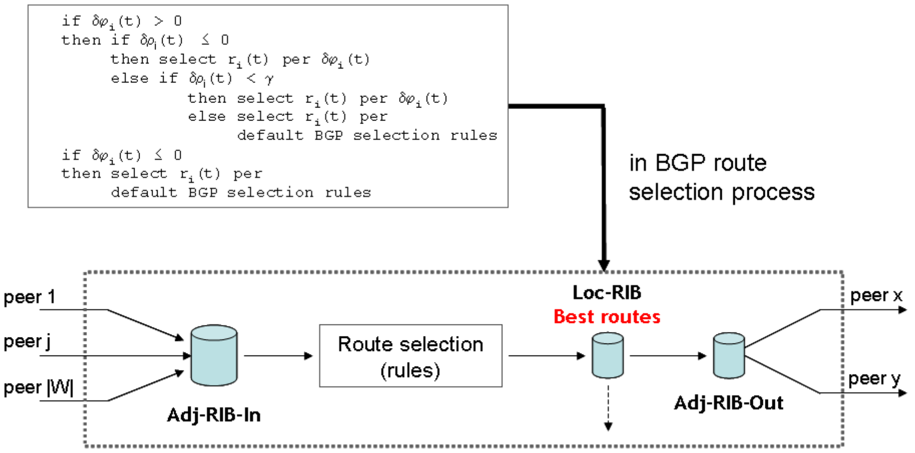
iv) Differential Stability-based decision criteria

The BGP route selection process can thus be enhanced by the stability-based decision criteria, following the differential stability metric defined here above. Using this additional rule the BGP route selection process would be driven by the rules detailed in Fig.1. The inclusion of this new route set of decision rules as part of the BGP route selection process is shown in Fig.2.

```
if $\delta\varphi_i(t) > 0$
then if $\delta\rho_i(t) \leq 0$
     then select $r_i(t)$ per $\delta\varphi_i(t)$
     else if $\delta\rho_i(t) < \gamma$
             then select $r_i(t)$ per $\delta\varphi_i(t)$
             else select $r_i(t)$ per
                    default BGP selection rules

if $\delta\varphi_i(t) \leq 0$
then select $r_i(t)$ per
     default BGP selection rules
```

Fig.1. Differential stability based decision criteria



```
if $\delta\varphi_i(t) > 0$
then if $\delta\rho_i(t) \leq 0$
     then select $r_i(t)$ per $\delta\varphi_i(t)$
     else if $\delta\rho_i(t) < \gamma$
             then select $r_i(t)$ per $\delta\varphi_i(t)$
             else select $r_i(t)$ per
                    default BGP selection rules
if $\delta\varphi_i(t) \leq 0$
then select $r_i(t)$ per
     default BGP selection rules
```

Fig.2. Differential stability based decision criteria in BGP route selection process

In this selection process, the positive integer parameter γ is determined by the increase of the multiplicative stretch considered as acceptable. Hence, the actual problem becomes to find a mean to actually determine (or at least estimate) the acceptable stretch increase of the routing path that would result from the application of the stability-based decision criteria. Past experiments dedicated to the measure of the BGP AS-path length have shown that even if the average length of AS-paths is relatively stable (about 4 to 5), a significant fraction of AS-paths has a length up to 10 [11]. From this perspective, if we assume that a 10% increase of the multiplicative stretch would be acceptable (resulting multiplicative stretch would be equal to 1.1 instead of 1.0), then routes with an average AS-path length increase of one (1) AS-hop would instead be selected. Note that this study does not evaluate the increase in memory consumption required to store the routes with longer AS-path attributes. Moreover, the application of the stability-based decision criterion prevents propagation of the routing updates churn resulting from the occurrence of a path exploration event when the following conditions are met i) the route corresponding to the next stable state is locally stored in the Adj_RIB_In and ii) this route corresponds to the most stable (next) route in the Adj_RIB_In. Indeed, if such event occurs, then the selection of a stable route becomes possible without delaying local convergence resulting from the exploration of all intermediate routing states (e.g., AS-paths of increasing length). Nevertheless, if the path exploration event also affects the route corresponding to the next state corresponding to the most stable next route, then selecting the AS-path that is the least topologically correlated[2] to the previous state provides the safest decision.

Importantly, the applicability of the stability-based decision criterion does not only depend on the point-value of the differential stability metric but also on its evolution over time. This means in practice that we have also to ensure that when the stability criteria are met at time t, and the corresponding selection rules are applied at time t, they also remain applicable at time t+1, and more generally at time t+Δt, where Δt >> 0. The reason stems as follows: at a given router once a route is selected at time t based on its stability properties, reverting unilaterally to the default BGP selection rules at time t+Δt can itself increase the instability induced by the concerned routes on its downstream routers.

Here again, our stability metrics provide a suitable method to estimate the deviation over time and the robustness of the selection process. Indeed, it suffices to notice that (even if it is impossible to locally anticipate all occurrence of BGP instability events before they occur) these metrics enable to determine over time the candidate replacement routes that are more stable compared to the set of possible alternative routes that do not show the same stability properties. When such alternative route does not exist, the exchange process of BGP routing updates between the local router and its downstream neighbors (with respect to the direction of propagation of the routing updates) requires enhancement in order to enable a smooth transition between the route selection rules. This mechanism performs as follows: anticipatively once no candidate replacement route is available for the route currently selected based on the stability criteria, that route is advertized to downstream neighbors together with the route that would be selected based on the default BGP selection rules. This process enables each downstream router to tune its decision process based on its

[2] Two AS_paths are topologically correlated if they share at least one common edge, i.e., an AS adjacency.

own selection rules for that route. Note that this process enables to advertize both routes, i.e., the one selected based on the stability criteria and the one selected based on the BGP default rules.

v) Potential effectiveness of the stability-based decision criteria

Even if selecting a more stable routing path could be considered as valuable from a routing level perspective, it does not necessarily imply that the corresponding forwarding path(s) would be itself more stable.

Henceforth, our first objective consists in determining if the dynamics of the Internet routing and forwarding system (through the analysis of routing and forwarding path instability) show different properties. If this assumption is verified then as one can not straightforwardly derive the one from the other; our second objective becomes to investigate the relationship between the stability of the forwarding path followed by the traffic and the corresponding routing path as selected by the path-vector routing protocol. For this purpose, we locally relate at the router level, the stability measurements carried on forwarding paths with the corresponding routing paths following the method developed in [8].

In the paper available in Annex 3, we provide an overview on prior work concerning the BGP routing system stability. We document the measurement and processing methodology together with the real datasets onto which these metrics have been applied. Finally, we report on the measurement results. Our analysis shows that the main cause of instability results from the forwarding plane (the dominant instability behavior is characterized by a majority of (FP_unstable,RP_stable) events). This observation further corroborates the assumption that the dynamic properties of the forwarding and the routing system are different. However, it can also be observed that a second order effect relates forwarding and routing path instability events. This observation provides first indication that a BGP route selection criteria path based on differential stability (see [7] [8]) is derived that can safely be taken into account as part of the BGP route selection process.

3.2 Path-vector routing

3.2.1 Route Verification

This currently ongoing work aims to investigate partial route verification in BGP and its relation with AS administrators' policies. Our work originates from the following two observations:

(1) BGP with/without route verification is/is not incentive-compatible.
(2) Route verification is extremely costly and, hence, impractical.

Still, incentive compatibility is an important objective since it will decrease the impact of policing in routing. So, we consider partial route verification of the following form: instead of checking whether the whole path declared by an AS is true, pick a few links in its path and verify that they indeed carry traffic from this AS. In its simplest form, such a scheme will check only one link per AS. Now, there are several issues related to whether such a (cheap) verification is effective. First, can it catch a lying AS? Well, it if it done deterministically, the answer is, in general, no: the AS can adapt is policy according to the verification and still manipulate BGP.

However, the use of randomness can improve this situation. By picking a few random links (e.g., one or two, given that paths are short in practice anyway) in the path declared by the AS, the lying agent can be caught with considerable probability. Then, a reasonable (and standard in the Economics literature) definition for the rationality of AS administrators is that it is an expected utility maximizer. Essentially, the AS will deviate from truth-telling only if such a policy will increase his gain. Of course, this requires the adoption of a convincing utility model which is not obvious by the definition of BGP (ASes declare preferences on paths as opposed to exact utilities for each of them). Here, we can assume utility models for each AS that are consistent to its ordinal preference. Based on such models (but without using the specific utilities for each AS), we can then design probabilistic verification schemes that are strategy-proof in expectation (i.e., the ASes have no incentive to deviate from BGP since their expected utility will not increase).

These ideas, in slightly different contexts, have been considered in two recent papers that appeared in ACM EC 2012 [9] [10]. The former defines the notion of probabilistic verification in mechanism design and presents positive and negative characterization and complexity results. The latter examines the role of underlying utilities in preference aggregation. Our current work on BGP is heavily influenced by these two works. Soon, we will have more concrete BGP-related results.

3.3 New routing paradigms

3.3.1 Components

As explained in Section 2, a complete protocol alternative to BGP is not

i) Locators assigned from Geometric space

Geometric routing refers to a class routing schemes that operate by assigning to nodes (virtual) coordinates in a metric space; these (virtual) coordinates are then used as addresses to perform point-to-point routing in this space. In this case, the routing paths whose destination is designated by a locator (logical spatial designator of an attachment point to the network topology) is derived from a geometric metric space, e.g., coordinate. Note this association can be either direct or indirect (requiring resolution).

The salient feature of geometric routing is that it relies on a locator space but this space is not equivalent to the usual topologically-dependent value space corresponding to network attachment points (like IPv4 assigned on an Ethernet interface). Indeed, coordinates can be inferred from the position of the node according to a reference space, in particular the hyperbolic space. Henceforth, automated address allocation can be considered by means of a coordinate inference procedure. However, no efficient technique (meaning technique not deteriorating other performance metrics such as the stretch) is currently known that would prevent the construction of a single and global coordinate space. Moreover, applicability of this locator-based addressing scheme for multicast groups (*,G) and (S,G) but also mobile nodes remains to be determined.

ii) Greedy routing

In greedy routing, each node performs distance computation for each packet based on local routing information it stores from local neighbor discovery.

To understand the term greedy we need to distinguish between routing paths corresponding to the logical concatenation of local decisions (without any associated local state) and the routing paths corresponding to the result of a distributed computation that is stored locally and to which a soft- or a permanent state is associated. In the former case, routing paths have no associated state whereas in the second case each routing path is associated to a local state. We refer to this distinction as "greedy routing" (also referred to as stateless routing) vs. "stateful routing".

To ensure that routing paths are loopfree, greedy routing relies on distance preserving embedding of the "observable" topology (being the actual router/POP topology). The embedding of undirected graph G = (V,E) in a metric space (X,d) : $G = (V,E) \rightarrow (X,d)$ is defined as a one-to-one mapping function μ: $V(G) \rightarrow X$. An embedding is distance preserving if $\forall$ s, t $\in$ V(G), s $\neq$ t, $d_G(s,t) \rightarrow d_X(\mu(s),\mu(t))$. A distance preserving embedding possesses the following property: $\forall$ s, t $\in$ V(G), s $\neq$ t, $\exists$ u $\in$ V(G) such that (s,u) $\in$ E(G), $d(\mu(s),\mu(t)) > d(\mu(u),\mu(t))$. Hence, $\forall$ s, t $\in$ V(G), s $\neq$ t, a distance decreasing path from μ(s) to μ(t) always exist. The path $(v_0(=s),v_1, \ldots, v_m(=t))$ is distance decreasing if $d(v_{i-1},v_m) > d(v_i,v_m)$, $\forall$ i. Moreover, an embedding of $G = (V,E) \rightarrow$ metric space (X,d) is said to be greedy iff greedy routing is always successful (i.e., a distance decreasing path can always be found in the embedded space X) [12]. In order words, each node of G is assigned the coordinates of the corresponding point of X and the distance between the points of X is the only information necessary for the route computation.

The fundamental challenges in greedy routing are to i) Find appropriate mapping function μ together with polynomial-time algorithm to embed V(G) in X so as to produce low stretch greedy routing paths using the metric d associated to that space, ii) Find procedure that does lead (in the worst case) to routing path stretch linear in n instead of poly-logarithmic in n (~log(n)), and iii) More importantly topology update such as node joining or leaving the network requires O(n) operations as the greedy property of the entire embedding can be invalidated. Current procedures requires knowledge of the full topology in advance (a link must be aware that it is "long"). Hence, one needs to reduce the communication cost and associated overhead without imposing drawbacks of centralized static routing scheme.

Nevertheless, the capability to build a set of local (per node) routing entries whose total memory size is proportional to the degree of each node (if we exclude the memory mobilized for storing the results of the operations for coordinate assignment) to forward traffic along distance decreasing paths is the main feature to be retained from this approach.

iii) Stochastic routing

The main idea underlying stochastic routing consists in i) performing routing decision by relying only on the local communication used by each node as the result of the minimization or maximization of (multiple) objective functions possibly subject to a set of constraints and ii) keeping accurate statistics on which routing decisions lead to optimize these functions, i.e., minimal delivery times. The salient feature of stochastic routing is its capability to account for uncertainty in routing information.

Stochastic routing is in the "hallways" since about 20 years and there is no scalable mean to take full benefit of the reinforcement learning algorithm embedded into each node. One possible way would consist in aggregating the state maintenance problem into an agent-based model where each agent would be responsible for a state sharing similar characteristics. Even each router/node would be viewed a single macro-agent [11] and the routing system as a Multi-Agent System (MAS) is viewed as a single macro-agent, joint action learner rapidly faces the curse of dimensionality, as both state and action spaces are tensor products of individual state and action space. Their size thus increases exponentially with the number of agents; only two agents are for instance considered in [11]. Moreover, this approach requires agents to somehow communicate instantly with each other (or at least access the state of the environment). While this hypothesis is unrealistic, the communication requirement can be relaxed by assuming that only some agents communicate with the others, or receive information from some master nodes.

3.3.2 Compact Multicast Routing

Dynamic compact multicast routing algorithms enable the construction of point-to-multipoint routing paths from any source to any set of destination nodes (or leaf nodes). The tree determined by a point-to-multipoint routing path is commonly referred to as a Multicast Distribution Tree (MDT) as it enables the distribution of multicast traffic from any source to any set of leaf nodes. By means of such dynamic routing scheme, MDTs can dynamically evolve according to the arrival of leaf-initiated join/leave requests. The routing algorithm creates and maintains the set of local routing states at each node part of the MDT. From this state, each nodes part of the MDT can derive the required entries to forward the multicast traffic received from a given source to its leaves.

In [15] we introduce a dynamic compact multicast routing algorithm that enables the construction of point-to-multipoint routing paths (referred to as Multicast Distribution Tree or MDT) for the distribution of multicast traffic from any source to any set of leaf nodes. An extended version of the compact multicast routing algorithm is further documented in Annex 4 (paper: "Design and Performance Analysis of Dynamic Compact Multicast Routing"). The novelty of the proposed algorithm relies on the information obtained locally and proportionally to the node degree instead of requiring knowledge of the global topology information (proportional to the network size). During the MDT construction, the routing information needed to reach a given multicast distribution tree is acquired by means of an incremental two-stage search process. This process, triggered whenever a node decides to join a given multicast source, starts with a local search covering the leaf node's neighborhood. If unsuccessful, the search is performed over the remaining unexplored topology (without requiring global knowledge of the current MDT). The returned information provides the upstream neighbor node along the least cost branching path to the MDT rooted at the selected multicast source node. The challenge consists thus here in limiting the communication cost, i.e., the number of messages exchanged during the search phase, while keeping an optimal stretch - memory space tradeoff.

i) Comparison with other Compact Multicast Routing schemes

AS far as our knowledge goes, prior work on compact multicast routing is, mainly concentrated around the routing schemes developed in the seminal paper authored by Abraham in 2009 [13]. One of the reasons we can advocate is that despite the amount of research work dedicated to compact unicast routing, current schemes are not yet able to efficiently cope with the dynamics of large scale networks. Therefore, running compact multicast routing independently of the underlying unicast routing system would be beneficial.

Compared to the Abraham compact multicast routing scheme [13], our name-independent compact multicast routing algorithm is also i) leaf-initiated since join requests are initiated by the leaf nodes; however, contrary to the Abraham scheme it operates without requiring prior local dissemination of the node set already part of the MDT or keeping specialized nodes informed about nodes that have joined the MDT, and ii) dynamic since requests are processed on-line as they arrive without re-computing and/or re-building the MDT. Moreover, our proposed algorithm is iii) distributed since transit nodes process homogeneously the incoming requests to derive the least cost branching path to the MDT without requiring any centralized processing by the root of the MDT or any specialized processing by means of pre-determined center nodes, and iv) independent of any underlying sparse cover construction grown from a set of center nodes (which induce node specialization driving the routing functionality): the local knowledge of the cost to direct neighbor nodes is sufficient for the proposed algorithm to properly operate. It is important to emphasize that the sparse cover underlying the Abraham scheme is constructed off-line and requires global knowledge of the network topology to properly operate.

ii) Comparison with current IP Multicast Routing schemes

This independence is the fundamental concept underlying multicast routing schemes such as Protocol Independent Multicast (PIM) [14]. Its variants for any-source multicast (PIM-SM) and single-source multicast (PIM-SSM) are the most commonly deployed routing protocols even if limited in scope (single carrier). Nevertheless, we also observe that the scaling problems experienced by these routing protocols and more generally all multicast routing approaches developed by the research community, remain largely unaddressed since so far. Indeed, multicast currently operates as an addressable IP overlay (Class D group addresses) on top of unicast routing topology, leaving up to an order of 100millions of multicast routing table entries. Hence, the need to enable point-to-multipoint routing paths (for bandwidth saving purposes) while keeping multicast addressing at the edges of the network and building shared but selective trees inside the network. In our approach, multicast forwarding relies on local port information only. Thus, the memory capacity savings comes from i) keeping 1:N relationship between network edge node and the number of multicast groups (N), and ii) local port-based addressing for the local processing of multicast traffic. Further, we argue that compact multicast routing by providing the best memory-space vs. stretch tradeoff, can possibly address the memory scaling challenges without requiring deployment of a compact unicast routing scheme.

BGPv4 has also been extended to support multicast discovery protocol. This extension relies on the multiprotocol BGP (MBGP) feature defined in RFC 2858. The multi-protocol capability of BGP enables multicast routing and to connect multicast topologies within and between BGP autonomous systems. In other words, multiprotocol BGP (MBGP) is an enhanced BGP that carries IP multicast routes. BGP carries two sets of routes, one set for unicast routing and one set for multicast routing. The routes associated with multicast routing are used by the Protocol Independent Multicast (PIM) to build data distribution trees. More recently, this feature has further been extended and BGPv4 can now also be used as multicast signaling protocol; hence, avoiding the use of PIM.

iii) Overview of our Compact Multicast Routing algorithm

The objective of the proposed compact multicast routing algorithm (referred to as PPC) is to minimize the routing table sizes of each node part of the

MDT at the expense of i) routing the packets on point-to-multipoint paths with relative small deviation compared to the optimal stretch obtained by the Steiner Tree (ST) algorithm, and ii) higher communication cost compared to the Shortest Path Tree (SPT) algorithm. For this purpose, the proposed algorithm reduces the local storage of routing information by keeping only direct neighbor-related entries rather than tree structures (as in ST) or network graph entries (as in both SPT and ST). In other terms, the novelty of the proposed algorithm is on requiring maintenance of only local topology information while providing the least cost next hop during the MDT construction. That is, our algorithm does not rely on the knowledge of the global topology information or involve the construction of global network structures such as sparse covers. The information needed to reach a given multicast source is acquired by means of a two-stage search process that returns the upstream node along the least cost branching path to the MDT sourced at s. This process is triggered whenever a node decides to join a given multicast source s, root of the MDT. After a node becomes member of an MDT, a multicast routing entry is dynamically created and stored in the local tree information base (TIB). From these routing table entries, multicast forwarding entries are locally instantiated.

As stated before, the reduction in memory space consumed by the routing table entries results however in higher communication cost compared to the reference algorithms, namely the SPT and the ST. Higher cost may hinder the applicability of our algorithm to large-scale topologies such as the Internet. Hence, to keep the communication cost as low as possible, the algorithm's search process is segmented into two different stages. The rationale is to put tighter limits on the node space by searching locally in the neighborhood (or vicinity) of the joining leaf node before searching globally. Indeed, the likelihood of finding a node of the MDT within a few hops distance from the joining leaf is high in large topologies (whose diameter is logarithmically proportional to its number of nodes) and this likelihood increases with the size of the MDT. Hence, we segment the search process by executing first a local search covering the leaf node's vicinity ball, and, if unsuccessful, by performing a global search over the remaining topology. By limiting the size (or order) of the vicinity ball while taking into account the degree of the nodes it comprises, one ensures an optimal communication cost. For this purpose, a variable path budget π_b is used to limit the distance travelled by leaf initiated requests to prevent costly (in terms of communication) local search or global search. Additionally, as the most costly searches are resulting from the initial set of leaf nodes joining the multicast traffic source, each source constructs a domain (referred to as source ball). When a request reaches the boundary of that domain it is directly routed to the source.

The proposed compact multicast routing algorithm is further documented in Annex 4 (paper: "Design and Performance Analysis of Dynamic Compact Multicast Routing"). This paper evaluates the performance of the proposed algorithm in terms of the stretch of the point-to-multipoint routing paths it produces, the size and the number of routing table entries, and the communication/messaging cost. Performances have been evaluated by simulation on synthetic power-law graphs (modeling the Internet topology) and the CAIDA map of the Internet topology comprising 32k nodes. It also compares its performance against legacy multicast routing algorithms (the Shortest Path Tree and the Steiner Tree algorithm). In that respect, the performance obtained with the proposed compact multicast routing scheme shows substantial gain in terms of the number of RT entries compared to the Steiner-Tree (ST) heuristic (minimum factor of 3,21 for sets of 4000 leaf nodes, i.e., 12,5% of the topology size) and the memory space required to store them. The stretch deterioration compared to the ST algorithms ranges between 8% and 3% (for multicast group size of 500 to 4000, respectively); thus, decreasing with increasing group sizes. The proposed two-phase search process -local search first covering the leaf's node vicinity, and if unsuccessful, a global search over the remaining topology- combined with the vicinity ball construction at the source node enables to keep the communication cost of the proposed algorithm within reasonable bounds compared to the reference Shortest Path Tree (SPT) scheme and sub-linearly proportional to the size of the leaf node set. Future work will determine if these promising performance results can still be verified for dynamic sequences of node join and node leave events and non-stationary topologies.

The comparison by the proposed algorithm and by the Abraham routing scheme as specified in [13] (for dynamic join only events) of their performance in terms of the stretch of the point-to-multipoint routing paths and the memory space required show that i) considering an aspect ratio[3] of 6 (and a network of 32k nodes the stretch of the Abraham scheme is about 3.5. Thus the stretch upper bound of the point-to-multipoint routing path produced by the Abraham scheme, even if universal (applicable to any graph), is about 3 times higher than the one produced by our scheme, ii) following its specification, the Abraham scheme requires a memory storage of about 700kbits per node for a tree comprising 4000 leaf nodes. For the same leaf set size, our compact multicast routing scheme requires about 1250kbits. These results seem to show that the proposed scheme provides a different stretch-memory tradeoff than the Abraham scheme noticing that the degradation in memory space is relatively limited.

[3] In this formula, the factor Δ is the aspect ratio defined as the ratio between max d(u,v) and min d(u,v), for any u, v $\in$ V.

4. Conclusion

It is relatively difficult to draw definitive conclusion concerning the potentiality of a specification that would provide a suitable alternative to the BGP routing protocol and that can be positioned as a genuine alternative routing paradigm. It is indeed not possible at this point in time, to infirm or to conform whether such alternative can ever be designed or not. What is clear though is that ***performance improvement is not a sufficient condition for migration*** to a new routing protocol (assuming that routing protocol would exist); moreover, ***functional preservation (if not improvement) is a necessary condition*** **to meet by any potential candidate. The alternative paradigms considered so far and the possible combination of their salient features together with the experiments to be conducted during the third year of the project may provide further evidences that such ultimate goal is achievable.**

Assuming such alternative would not be achievable, the efforts conducted in the context of the Task 2.2 have also led to propose genuine improvements to the BGP routing protocol (even if some are incremental) but also propose foundational building blocks to path-vector routing such as the inception of a routing information exchange process to replace the base push-model of BGP and a partial route verification process.

References

[1] C.Villamizar, R.Chandra, and R.Govindan, BGP Route Flap Damping, Internet Engineering Task Force (IETF), RFC 2439, November 1998.

[2] Z.M.Mao, R.Govindan, G.Varghese, and R.Katz, Route Flap Damping Exacerbates Internet Routing Convergence, Proc. of ACM SIGCOMM 2002, Pittsburgh (PA), USA, August 2002.

[3] R.Bush, T.Griffin, and Z.M.Mao, Route flap damping harmful?, NANOG-26, 28 October 2002.

[4] J.Chandrashekar, Z.Duan, Z.-L.Zhang, and J.Krasky, Limiting path exploration in BGP, Proc. of IEEE INFOCOM 2005, Miami (FL), USA, March 2005.

[5] D.Pei, M.Azuma, D.Massey, and L.Zhang, BGP-RCN: improving BGP convergence through root cause notification, Computer Networks, ISDN Syst. vol. 48, no.2, pp.175-194, June 2005.

[6] G.Huston, M.Rossi, and G.Armitage, A Technique for Reducing BGP Update Announcements through Path Exploration Damping, IEEE Journal on Selected Areas in Communications (JSAC), vol.28, no.8, October 2010.

[7] D.Papadimitriou, A.Cabellos, and F.Coras, Path-vector Routing Stability Analysis, Proc.13th Workshop on MAthematical Performance Modeling and Analysis, ACM SIGMETRICS, San Jose (CA), USA, June 2011.

[8] D.Papadimitriou, A.Cabellos, and F.Coras, Stability metrics and criteria for path-vector routing, To appear in Proc. of IEEE International Conference on Computing, Networking and Communication (ICNC) 2013, San Diego (CA), USA, January 2013.

[9] I.Caragiannis, E.Elkind, M.Szegedy, and L. Yu. Mechanism design: from partial to probabilistic verification. In Proceedings of the 13th ACM Conference on Electronic Commerce (EC 12), pp. 266-283, 2012.

[10] C.Boutilier, I.Caragiannis, S.Haber, T.Lu, A.D.Procaccia, and O.Sheffet, Optimal social choice functions: a utilitarian view, In Proceedings of the 13th ACM Conference on Electronic Commerce (EC 12), pp. 197-214, 2012.

[11] M.L.Littman. Markov games as a framework for multi-agent reinforcement learning. In Proc. 11th ICML, pages 157–163. Morgan Kaufmann, 1994.

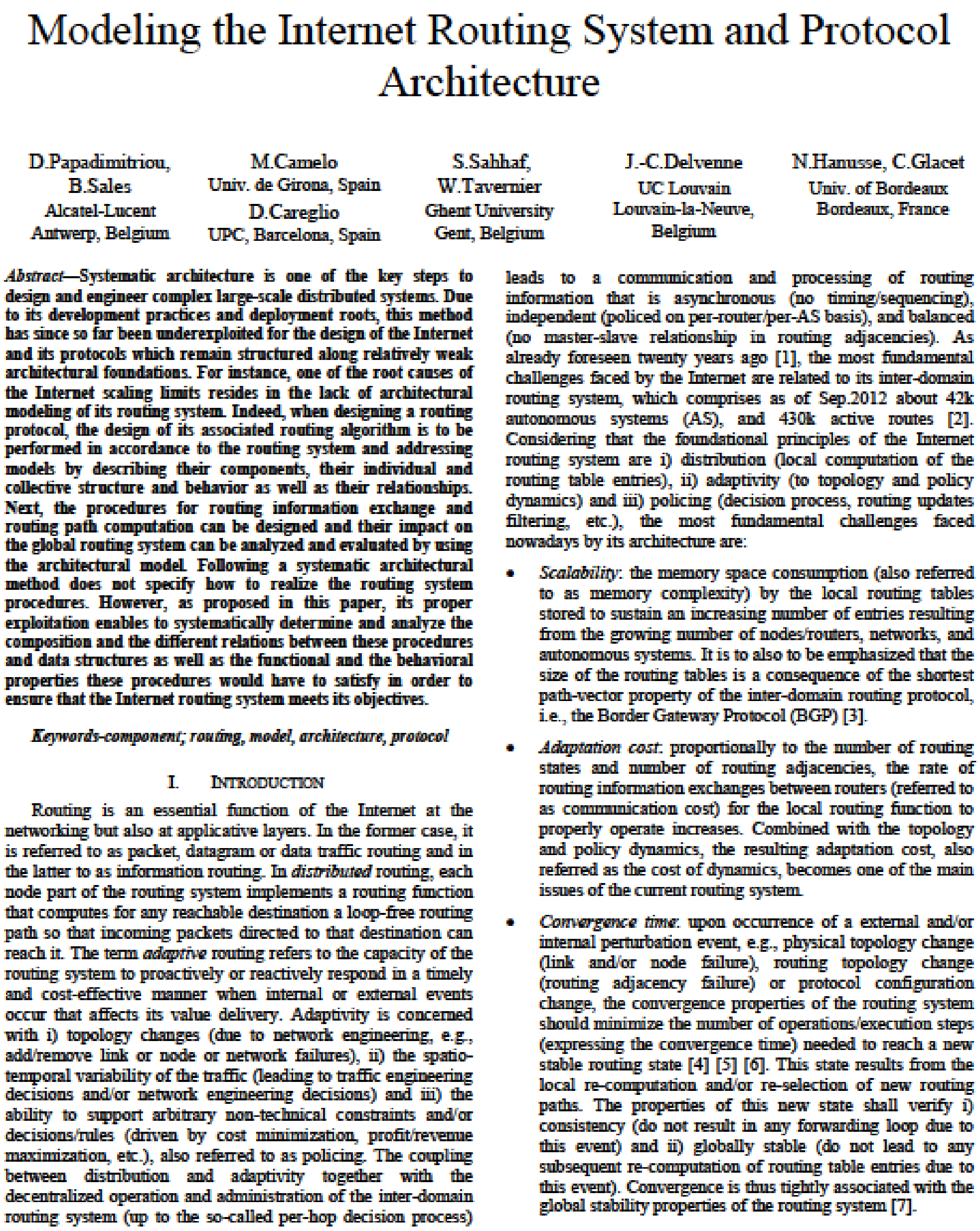
[12] C.Papadimitriou and D.Ratajczak, "On a conjecture related to geometric routing," Theoretical Computer Science, vol. 344, no. 1, pp. 3–14, 2005.

[13] I.Abraham, D.Malkhi, and D.Ratajczak, Compact multicast routing, Proc. 23rd Int. Symp. DISC'09, Elche, Spain, pp.364–378, Sep.2009.

[14] B.Fenner, et.al., Protocol Independent Multicast - Sparse Mode (PIM-SM), Internet Engineering Task Force (IETF), RFC 4601, Aug.2006.

[15] P.Pedroso, D.Papadimitriou, D.Careglio, "Dynamic compact multicast routing on power-law graphs", 54th IEEE Globecom 2011. Houston, TX, USA, Dec. 2011

Annex 1: Paper "Modeling the Internet Routing System and Protocol Architecture"



Modeling the Internet Routing System and Protocol Architecture

D.Papadimitriou, B.Sales
Alcatel-Lucent
Antwerp, Belgium

M.Camelo
Univ. de Girona, Spain
D.Careglio
UPC, Barcelona, Spain

S.Sahhaf, W.Tavernier
Ghent University
Gent, Belgium

J.-C.Delvenne
UC Louvain
Louvain-la-Neuve, Belgium

N.Hanusse, C.Glacet
Univ. of Bordeaux
Bordeaux, France

***Abstract*—Systematic architecture is one of the key steps to design and engineer complex large-scale distributed systems. Due to its development practices and deployment roots, this method has since so far been underexploited for the design of the Internet and its protocols which remain structured along relatively weak architectural foundations. For instance, one of the root causes of the Internet scaling limits resides in the lack of architectural modeling of its routing system. Indeed, when designing a routing protocol, the design of its associated routing algorithm is to be performed in accordance to the routing system and addressing models by describing their components, their individual and collective structure and behavior as well as their relationships. Next, the procedures for routing information exchange and routing path computation can be designed and their impact on the global routing system can be analyzed and evaluated by using the architectural model. Following a systematic architectural method does not specify how to realize the routing system procedures. However, as proposed in this paper, its proper exploitation enables to systematically determine and analyze the composition and the different relations between these procedures and data structures as well as the functional and the behavioral properties these procedures would have to satisfy in order to ensure that the Internet routing system meets its objectives.**

Keywords-component; routing, model, architecture, protocol

I. Introduction

Routing is an essential function of the Internet at the networking but also at applicative layers. In the former case, it is referred to as packet, datagram or data traffic routing and in the latter to as information routing. In *distributed* routing, each node part of the routing system implements a routing function that computes for any reachable destination a loop-free routing path so that incoming packets directed to that destination can reach it. The term *adaptive* routing refers to the capacity of the routing system to proactively or reactively respond in a timely and cost-effective manner when internal or external events occur that affects its value delivery. Adaptivity is concerned with i) topology changes (due to network engineering, e.g., add/remove link or node or network failures), ii) the spatio-temporal variability of the traffic (leading to traffic engineering decisions and/or network engineering decisions) and iii) the ability to support arbitrary non-technical constraints and/or decisions/rules (driven by cost minimization, profit/revenue maximization, etc.), also referred to as policing. The coupling between distribution and adaptivity together with the decentralized operation and administration of the inter-domain routing system (up to the so-called per-hop decision process) leads to a communication and processing of routing information that is asynchronous (no timing/sequencing), independent (policed on per-router/per-AS basis), and balanced (no master-slave relationship in routing adjacencies). As already foreseen twenty years ago [1], the most fundamental challenges faced by the Internet are related to its inter-domain routing system, which comprises as of Sep.2012 about 42k autonomous systems (AS), and 430k active routes [2]. Considering that the foundational principles of the Internet routing system are i) distribution (local computation of the routing table entries), ii) adaptivity (to topology and policy dynamics) and iii) policing (decision process, routing updates filtering, etc.), the most fundamental challenges faced nowadays by its architecture are:

- *Scalability*: the memory space consumption (also referred to as memory complexity) by the local routing tables stored to sustain an increasing number of entries resulting from the growing number of nodes/routers, networks, and autonomous systems. It is to also to be emphasized that the size of the routing tables is a consequence of the shortest path-vector property of the inter-domain routing protocol, i.e., the Border Gateway Protocol (BGP) [3].
- *Adaptation cost*: proportionally to the number of routing states and number of routing adjacencies, the rate of routing information exchanges between routers (referred to as communication cost) for the local routing function to properly operate increases. Combined with the topology and policy dynamics, the resulting adaptation cost, also referred as the cost of dynamics, becomes one of the main issues of the current routing system.
- *Convergence time*: upon occurrence of a external and/or internal perturbation event, e.g., physical topology change (link and/or node failure), routing topology change (routing adjacency failure) or protocol configuration change, the convergence properties of the routing system should minimize the number of operations/execution steps (expressing the convergence time) needed to reach a new stable routing state [4] [5] [6]. This state results from the local re-computation and/or re-selection of new routing paths. The properties of this new state shall verify i) consistency (do not result in any forwarding loop due to this event) and ii) globally stable (do not lead to any subsequent re-computation of routing table entries due to this event). Convergence is thus tightly associated with the global stability properties of the routing system [7].

- *Stability*: the individual local routing states and associated routing path should remain (at least marginally) stable upon occurrence of perturbation resulting i) the exploration of the routing state space (compared to the BGP uninformed path exploration intrinsic to shortest-path vector algorithm) and ii) the routing policies interactions (compared to BGP routing policy interactions that can lead to non-deterministic and unintended but stable routing states, and "dispute wheels", i.e., non-deterministic and unintended but unstable states) [8] [9].

These challenges result from i) the increasing number of routing entries and thus routing states amplified by the design and usage of the addressing system (including prefix de-aggregation practices for traffic engineering purposes, and site multi-homing), ii) the short-term topology and policy dynamics and iii) the longer-term topology evolution (increasing meshedness). Their combination together with the intrinsic limits of the BGP architecture and its underlying properties leads to a very complex problem.

Alongside, one of the main root causes of the absence of suitable alternative to BGP resides in the lack of architectural modeling of the global routing system when designing a routing protocol and its associated routing algorithm(s). Indeed, such design is to be performed in accordance to the routing system and addressing model describing their components and relationships (and not independently). Next, the procedures for routing information exchange and routing path computation can be specified and their impact on the global routing system can be analyzed and evaluated by using the architectural model. Following a systematic architectural method does not specify how to implement the routing procedures and data structures themselves. However, the proper exploitation of this method enables to systematically determine and analyze the composition and the different relations between these procedures and data structures as well as the functional and the behavioral properties these procedures would have to satisfy in order to ensure that the Internet routing system meets its objectives. When the routing system is not properly modeled, the impact of these design choices on the global routing system is almost impossible to evaluate beforehand making any improvement a trial-and-error experiment. Moreover, experience shows that without well-defined routing system architecture, adding/removing or replacing routing functionality increases its architectural complexity; In order to address these challenges altogether, we propose in this paper, to model the routing system by identifying its functional components, their relationships, and their spatio-temporal distribution (functional model) together with the information properties, operations and relationships (information model).

This paper is structured as follows. Section II documents prior work in Internet routing system architecture and protocols. Section III motivates the approach proposed in this paper. In Section IV, we detail the proposed method that comprises the specification of a functional and informational model. Subsequently, Section V details the both models and Section VI illustrates their application to the BGP routing system. Finally, Section VII concludes this paper and proposes future work following the proposed architectural model.

II. Prior Work

Prominent research efforts have been conducted over last decades to address the challenges related to the Internet routing system. These efforts can be classified as follows: i) incremental improvements to BGP, ii) new class of path-based routing protocols, and iii) new routing paradigms.

A. Incremental improvements to BGP

To avoid or mitigate some of the well-known BGP path-vector routing limitations many incremental improvements to BGP have been proposed in order to expectedly reduce or bound the performance degradation of the Internet routing system. These include mechanisms and techniques to i) shorten routing update interval to accelerate routing table entries convergence [8], ii) provide route flap damping to prevent or limit sustained route oscillations that could potentially put an undue processing load on BGP [10] [11], iii) add forward edge sequence number to annotate the AS-Paths with additional "path dependency" information (enhanced path-vector algorithm) [12], iv) explicitly indicate the AS-Path dependency and failure root cause/location information in BGP routing updates to mitigate path exploration effects [13], or v) include multiple AS-Path per destination to improve fault-tolerance by increasing AS-Path diversity [14]. Over time, some of these ad-hoc improvements have permitted to limit (up to a certain extend) the performance degradation of the Internet routing system. For others, it was subsequently shown that they induce detrimental effects to the routing system [11]. However, none of them improves the intrinsic limitations of the path-vector routing protocol architecture and algorithm impacting the scalability (stretch-1 routing paths) and the convergence (due to path exploration) properties of the routing system. Moreover, these improvements tend also to increase the complexity of the BGP architecture, measured in terms of the number of functional components and the number of relationships among them.

Other enhanced versions of BGP shall also be mentioned: in particular those that extend the security mechanisms against threats that can arise at various levels. A threat is defined per [RFC4593] as a potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm. Limitations of current BGP session security mechanisms (between BGP peers) include: i) The use of static keys, which tend to be changed infrequently, and often not at all and makes long term brute force attacks feasible; moreover, as keys are typically chosen by humans, and expressed in ASCII; the entropy is typically small, making the keys easier to determine, ii) The key change process needs to be coordinated between both sides of the BGP session, making this an operationally expensive exercise, iii) The dependence on the MD5 algorithm, which brings two problems: MD5 is not considered strong enough for the future as documented in [RFC4278], iv) The security architectures should also allow a choice of algorithms, to have an alternative in case serious vulnerabilities are discovered in an algorithm, and v) When confidentiality of BGP routing information is required can only be achieved today by securing the BGP session with IPsec. It is well-accepted that in order to improve the situation the following extensions would be valuable

- Use certification objects within this secure routing architecture for supporting the distribution of authorization and the authentication of the originated routing information. Indeed, the basic security questions that can be posed regarding routing information are whether the originating Autonomous System (AS) is authorized to advertise an address prefix by the holder of that prefix, whether the originating AS is accurately identified by the originating AS Number (ASN) in the advertisement, and the validity of both the address prefix and the ASN.
- Related to the previous point, one of the main targets is the level of trust than can be ascribed to attributes of a route object in terms of their authenticity, including consideration of the AS Path attribute
- Resolve the security limitations for single BGP sessions, i.e., the connection between two BGP peers, implies i) to ensure unicity of BGP speaker Identity, ii) to support means for BGP peer authentication, iii) to provide methods to ensure integrity of routing message exchanged between speakers, iv) to have mechanisms to encrypt BGP messages in transit (so as to ensure confidentiality), v) to detect and protect against anti-replay attacks (methods to detect and prevent replay of BGP routing messages), and vi) to protect the BGP session against denial of service attacks, targeting the availability of the BGP session.

Finally let's also mention extension to cope specifically with global routing stability by means of route verification. As demonstrated in [12] to achieve incentive compatibility of best-reply BGP dynamics requires combining this global condition (Route Verification) together with the "No Dispute Wheels" sufficient condition to guarantee stability. However, all known conditions, including the "No Dispute Wheels" condition, for global stability are sufficient but not necessary conditions whereas checking them is an NP-hard problem and enforcing them requires a global deployment of an additional mechanism.

B. *New Path-based Routing Protocols*

These protocols still rely on the notion of path as the central information unit; however, they combine it with mechanism enabling flexibility up to the senders that can be part of the route selection process. This class comprises protocols processing of path segments (instead of end-to-end paths) as proposed in Pathlet routing [15]. Pathlet is a source routing over a virtual topology scheme that relies on a representation of the Internet as a virtual topology independent of the physical topology. It uses two building blocks: i) a vnode is a virtual node that represents arbitrary granularities, such as an entire autonomous system (AS), a geographical region, or a class of policies, ii) a pathlet is a fragment of a path: a sequence of vnodes along which the originating AS is willing to route. Route computation is shifted to the edges: senders concatenate their selection of pathlets into an end-to-end source route represented as a list of identifiers in the packet header. Examples of pathlet routing's flexible routing policies include the emulation of any routing policy supported by a number of other protocols: BGP, loose and strict source routing, and recent multipath proposals such as MIRO [16], and NIRA [17]. Routing policies that are "local", in that they are a function only of the ingress and egress points in a network, can be represented using very small forwarding tables and lead to many choices of routes for senders—potentially an exponentially large number of path choices. Pathlet routing does not impose a global requirement on what style of policy is used, but rather allows multiple radically different styles to coexist. For example, one part of the Internet could use permissive policies that allow senders to choose any route while another part could use traditional restrictive BGP-style policies.

The NIRA (new Internet Routing Architecture) protocol functionality is similar to Pathlet routing. NIRA also gives a user the ability to choose the sequence of providers his packets take. NIRA intends to address broad range of issues, including practical provider compensation, route discovery, route representation, fast route fail-over, and security. NIRA supports user choice without running a global link-state routing protocol. It breaks an end-to-end route into a sender part and a receiver part and uses address assignment to represent each part. A user can specify a route with only a source and a destination address, and switch routes by switching addresses.

Finally MIRO (Multipath Inter-domain Routing), as its name indicates is a multi-path inter-domain routing protocol. It offers flexibility, while giving transit domains control over the flow of traffic through their infrastructure and avoiding state explosion in disseminating reachability information. In MIRO, routers learn default routes through the existing BGP protocol, and arbitrary pairs of domains can negotiate the use of additional paths (bound to tunnels in the forwarding plane) tailored to their special needs. MIRO retains the some of simplicity of BGP for most traffic paths, and remains backwards compatible with BGP to allow for incremental deployability.

All these protocols show actually similar issues. There is no path enforcement and no verification mechanism in addition to the lack of accountability; hence, the source has no mean to verify that the packet has actually followed the selected sequence and the receiver has no idea what path was followed. For the protocols operating end-to-end such as NIRA switching from one path to another (as traffic engineering is left up to the sender) may suffer from delays. The downside of the Pathlet approach is the complexity of creating policies that depend on prefixes of the path. Moreover, in Pathlet, the proposed dissemination protocol is not designed generically enough to efficiently support emulation of NIRA and MIRO. Hence, this observation questions the capacity of the dissemination protocol to sustain additional extensions in the future.

C. *New Routing Paradigms*

Investigation of new routing paradigms to address altogether the challenges listed in Section.I led also to new routing paradigms. The resulting routing schemes aim at providing a competitive tradeoff between the memory space required to sustain routing table size scaling and the computational resource required to support routing exchange dynamics while maintaining a low-stretch increase of the produced routing paths. Among them, compact routing, geometric routing, and stochastic routing have attracted over last decade main attention from the scientific and technical community.

- *Compact routing* aims to find the best tradeoff between the memory-space required to store the routing table (RT) entries at each node and the stretch factor increase on the routing paths it produces. The key idea of compact routing algorithm is to make routing table sizes compact by omitting "some" network topology details such that resulting path length increase stays small (bounded). Such routing schemes have been extensively studied following the model developed in [15]. Since then, in accordance to the distinction between labeled (nodes are named by polylogarithmic size labels encoding topological information) and name-independent (node names are topologically independent) compact routing schemes have been designed, notably the so-called AGMNT name-independent compact routing scheme [16]. This scheme is centralized and static (instead of being distributed and dynamic); it is thus by definition inapplicable for the Internet [17].

- *Geometric routing* refers to a class routing schemes that operate by assigning to nodes (virtual) coordinates in a metric space; these (virtual) coordinates are then used as addresses to perform point-to-point routing in this space. The salient feature of geometric routing is that it builds a set of local routing entries whose memory size is proportional to the degree of each node (if we exclude the memory mobilized for storing the results of the operations for coordinate assignment). More recent advances in (distributed) geometric routing take benefit of the intrinsic properties of the Internet topology such as its curvature [18] [19] and its δ-hyperbolicity [20], which to some extent measures its deviation from tree-likeness [21].

The routing path length increase and the adaptation cost of these routing schemes are hardly compatible with the requirements expected from an alternative to BGP [20]. In this context, specifying a generic distributed and dynamic routing model would expectedly help designing a compact or geometric routing protocol adapted to the Internet.

- *Stochastic routing* and its multiple variants [22] rely on reinforcement learning algorithm embedded into each node. The main idea underlying stochastic routing consists in i) performing routing decision by relying only on the local commmunication used by each node as the result of the minimization or maximization of (multiple) objective functions possibly subject to a set of constraints and ii) keeping accurate statistics on which routing decisions lead to optimize these functions, i.e., minimal delivery times. The salient feature of stochastic routing is its capability to account for uncertainty in routing information.

The design of these routing schemes tends to follow (at least since so far) the exact same approach as the one pursued by BGP. This statement is corroborated by the following observations: i) the routing algorithm still exclusively determines the behavior of the routing system whereas a proper method would assume that the routing system architecture (which comprises a non-local information acquisition function) determines which class of algorithms produces the needed output from the available input (and under which conditions), ii) certain performance objectives are verified by the routing algorithm but without accounting for their dependency on the spatial and *temporal* properties of the information/input and running conditions (e.g., memory space consumption is minimized in stationary conditions up to the point that adaptivity cost and convergence time objectives become unachievable), and iii) the functional distinction between the routing information acquisition function being either explicit (push/pull) or implicit (local inference) and the routing path computation function is often neglected. On the other hand, little work has been realized since so far in terms of architectural modeling of the Internet routing system with the purpose of deriving alternative routing schemes (and subsequently routing protocols). This situation has led to a deadlock in terms of routing research since approaching the problem space requires the design of routing algorithm(s) and protocol but also the specification of the routing architecture that couples both information and functional model. We argue that failing to work simultaneously and in symbiosis with these three dimensions altogether, explains for a large part the reason why no suitable alternative to the BGP-based routing system has been proposed since so far but also why no adequate improvements to BGP have been designed since so far. The other reason is the lack of analytical model translating the behavior (in particular, the spatio-temporal properties) of the entities inducing network dynamics but on which the behavior of the routing system and protocol depends.

III. Motivations

As this paper focuses on the architecture of the routing system, defining the term system architecture is crucial. Many definitions of system architecture have been formulated over time. Our definition combines elements from D.E.Perry and A.L.Wolf in [21], D.Garlan and D.E.Perry in [22] and G.Booch in [23]. We define the term *system architecture* as a set of functions, states, and objects/information (referred to as "elements") together with their behavior, their structure (composition, relationships and interactions) and their spatio-temporal distribution. The latter spatio-temporal distribution of the above-referenced elements is characteristic of distributed systems. The specification of these architectural elements comprises the functional model, the information/ object model and the state model, respectively. These models are further defined and used as part of the method proposed in Section.IV.

At this stage, one may question the relevance of performing such exercise, knowing that the Internet architecture itself relies on relatively weak foundations and its properties are heavily determined by its protocols. Historically, the design choices of these protocols were mainly driven by computational constraints (memory and CPU), per-AS decentralized control and decision as well as organic deployment; thus, far away from genuine architectural considerations. The main motivations for taking a systematic approach in specifying the architecture of the routing system can be summarized as follows:

- Specify a common architectural baseline that complements the routing algorithm design by a holistic design of the routing protocol (and their components) and that answers questions such as whether a common routing architecture can (partially) cover existing and new routing models/schemes outlined in Section II; and at which level of the routing system architecture specification do we need to introduce per-routing scheme customization and/or specialization;

- Determine which routing (sub-)functions are currently under-specified or for some inadequately specified but also which routing (sub-)functions can be replaced, added or even removed from the routing system architecture;
- Define a common description "language" together with the representation of the different interactions characterizing the Internet routing system to prevent misinterpretations among the different dimensions but also actors involved in the design of the routing system architecture;
- Specify the model of the routing system architecture up to the level appropriate for routing protocol engineering (including its various control interfaces) and in turn enable modular software development that prevents duplicates, e.g., several BGP functionality can as of today be realized by the composition of different procedures;
- Provide a functional analysis grid to compare different routing protocols and their associated components obtained from various design choices that are possible when specifying these protocols.

IV. Method

As determined by the definition of system architecture provided in Section III, the proposed method consists in specifying two complementary models: the functional model and the information model.

A. Functional Model

The basic idea underlying functional modeling is the following: the system is viewed as computing a function or, more generally, solving an information processing problem. Functional modeling proceeds by systematically describing the automated processing that a complex system must perform to transform available inputs to the desired outputs. For this purpose, functional modeling assumes that such processing can be explained by iteratively decomposing the corresponding complex function into a set of simpler (sub-)functions that are computed by an organized sub-system up to the level of atomic functions. The expectation being that when performing such decomposition, the resulting sub-functions taken individually will be simpler than the original function.

1) Definition

A functional model determines a systematic decomposition of the (routing) system by defining its functional design, its inputs/outputs, and its various interfaces. This modeling technique enables to systematically describe the automated processing that the routing system must perform to transform available inputs to the desired outputs. The underlying idea is the following: the routing system is viewed as a distributed computing function or, more generally, as solving a routing information processing problem. The processing performed by the routing system can be explained by iteratively decomposing the more complex top-level routing function (or functional area) into a set of simpler functions (or sub-functions) each computed by an organized sub-system. This decomposition is performed up to the level of atomic functions, i.e., as their name indicates atomic functions are functions that cannot by definition be further decomposed.

2) Approach

In the early design phase, functional modeling refers to a methodology part of the architectural specification process aiming at systematically identifying, describing, and relating the functions a system must perform (thus the functions that need to be included in the system) in order to meet its objectives. Functional modeling does not address how these functions will be performed or implemented but instead deals with i) *identification*: the specification of the top-level functions that need to be performed by the system and their decomposition into sub-functions together with the definition of their inputs/outputs, and their various interfaces, ii) *spatial distribution*: where theses functions need to be performed (space); iii) *temporal distribution*: how often they need to be performed (time); iv) *operation*: under which operational context and environmental conditions .

There are four elements to be addressed by the functional modeling approach: i) the *hierarchical decomposition* of the functions starting from the top-level function (or top-level functional area) of the system. The top-level function is partitioned into a set of sub-functions that use the same inputs and produce the same outputs as the top-level function. Each of these sub-functions can then be partitioned further, with the decomposition process continuing as often as it is useful or up to the atomic level, ii) the decomposition in *functional blocks* by means of Functional Flow Block Diagrams (FFBD) [24], that represent the information flow among the functions within any portion of the hierarchical decomposition. As the first and subsequent functional decomposition levels are examined, it is common for one function to produce outputs that are not useful outside the boundaries of the system. These outputs are needed by other functions in order to produce the needed and expected external outputs, iii) the *processing instructions* that contain the needed information for the functions to transform the inputs to the outputs, and iv) the *control flow* (including the triggers) that sequences the termination and activation of the functions so that the process is both efficient and effective. Section V.A details the hierarchical decomposition of the routing functional model, further details concerning the functional block/FFBD decomposition together with instruction processing and control flow can be found in [27].

B. Information Model

1) Definition

Information/object model provides a representation of the concepts, relationships, properties, constraints, rules, and operations to specify the information semantics for a given application domain/area. Hence, such model is also referred to as semantic data model or conceptual data model. There are different methods for developing an information model. Among them the Entity-Relationship (E-R) model expresses in terms of entities (represented by rectangles), the relationships (represented by diamonds) among these entities and the attributes (represented by ovals) of both entities and relationships [25]. The ultimate goal of applying such model is to capture as much of the meaning of the information (semantic) as possible so as to obtain a better design that is scalable and easier to maintain.

2) Approach

Information modeling enables to represent the properties, relationships, constraints, rules, and operations in order to formally specify the information semantics for a given application domain/area. In its original form, the Entity-Relationship (E-R) modeling technique does not support specialization/generalization abstractions (also termed hierarchies). Hence, the E-R modeling technique has been subsequently extended to include additional concepts: set-subset relationships (sub-classes and super-classes), specialization/ generalization, categories, and attribute(s) inheritance. The resulting model, referred to as Enhanced E-R or Extended E-R model is commonly used to model applications more completely and accurately (if needed). Hence, we will make use of this modeling method following the objectives of this paper to specify the information model associated to the routing system providing that the resulting information model i) encompasses most of existing routing protocols (e.g., path vector routing, distance-vector routing, link-state routing) but also new routing protocols (e.g., compact routing, geometric routing and their variants); ii) remains flexible enough to facilitate different interfaces and accommodate other data models that may have a different logic. Different relationships between the entities may determine different interfaces; and iii) reduces the architectural complexity which measures the complexity of the architecture proportionally to its number of components (in the present case objects/entities) and the interactions (relationships) between these components.

V. Models Specification

This section specifies the functional model (by means of the hierarchical decomposition of the routing functional area) and information model (by means of the E-R model).

A. Hierarchical Decomposition of the Routing function

Following the definition provided by [26], a function is a transformation process that changes inputs into outputs. The Routing or Route functional area is defined as the local process of determining (computing) and deciding (selecting) a loop-free routing path for any destination node such that the traffic directed to each destination will reach its destination. From this definition, the routing function can be modeled by a single, top-level area or top-level function that can be decomposed into a hierarchy of sub-areas or sub-functions. Fig.1 depicts the hierarchical decomposition up to the fourth level (further decomposition being documented in [27]). The following sub-sections describe the routing function and its sub-functions following the hierarchical decomposition of the routing functional area in 1.Discover function, 2.Pre-process function, 3.Produce routing path function, 4. Produce routing table entry function, and 5.Associated functions (not shown in Fig.1).

1) Discover function

A common approach for decomposing the Discover function (also referred to as information acquisition function) is to segment this function with respect to specialization following the processing and exchange of two main classes of information, i.e., routing and topology information:

- *Routing information discovery* where routing information refers to i) distances (or information to derive these distances) together with their attributes, and/or ii) routes or route segments (or information to derive these routes) together with their attributes. Note that routing information can also be obtained (explicitly) or derived (implicitly) from the spatial and the temporal properties of the path(s) followed by the traffic flows.

- *Topology information discovery* where topology information refers to all information related to i) local and remote interfaces, incident links, nodes adjacent to incident links together with their attributes, and/or ii) non-local information including remote links and (abstract) nodes together with their attributes.

Both discovery functions can be further decomposed into Operate Discovered Information and Exchanged Discovered Information function.

a) Discover Routing Information function

The Discover Routing Information function is structured as follows:

- *Operate routing information* sub-function which sub-divides into: i) Create routing information entries in the Routing Information Base (RIB), ii) Update routing information entries in the RIB, and iii) Control routing information entries of the RIB.

- *Exchange routing information* sub-function which sub-divides into: i) Push routing information which includes disseminate routing information to neighbors (local), disseminate routing information network-wide (non-local) but also filtering/selection of the disseminated/acquired routing information, ii) Pull routing information which includes query/request from neighbors (local), query/request from network (non-local) but also filtering/selection of the pulled/acquired information, iii) Trigger and control exchanges (push/pull) of routing information, and iv) Structure routing information exchanged or to be exchanged (syntax function).

The distinction between local (or neighbor) and network (or non-local) discovery is commonly performed and applies to routing information. The local discovery function (also referred to as neighbor discovery function) enables the acquisition/ dissemination of knowledge about the local environment (neighborhood) to local entities including local and remote interfaces (and their properties), incident links (and their properties), and nodes adjacent to incident links (and their properties). On the other hand, the remote/non-local discovery function (also referred to as network discovery function) enables the acquisition/ dissemination of knowledge about the non-local environment from/to remote entities including remote links/nodes, paths and/or distances to reachable destinations.

b) Discover Topology Information function

The Discover Topology Information function is structured as follows:

- *Operate topology information* sub-function which subdivides into: i) Create topology information entries in Topology Information Base (TIB), ii) Update topology information entries in the TIB, iii) Control topology information entries of the TIB.
- *Exchange topology information* sub-function which subdivides into: i) Push topology information which includes disseminate topology information to neighbors (local), disseminate topology information network-wide (non-local) but also the selection of the disseminated/ acquired topology information, ii) Pull topology information which includes query/request from neighbors (local), query/request from network (non-local) but also selection of the pulled/acquired topology information, iii) Trigger and control exchanges (push/pull) of topology information, and iv) Structure topology information exchanged or to be exchanged (syntax function).

As for the Discover Routing Information, the distinction between local/neighbor and network/non-local discovery is commonly performed and applies also to topology information.

2) Pre-Process function

Pre-processing consists in structuring and/or analyzing the topology and/or routing information using a combination of the following operations:

- *Embedding/mapping*: given metric spaces (X,dX) and (Y,dY), where X and Y are spaces, and dX and dY are distance functions, a mapping function $\mu: X \rightarrow Y$, $x \rightarrow y=\mu(x)$ is called an embedding. An embedding is called distance-preserving if $x, y \in X$, $dX(x,y) = dY(\mu(x),\mu(y))$.
- *Composition*: this function combines topology and/or routing information so as to build more complex topology and/or routing information (called structures).
- *Mining*: includes all procedures enabling to automatically find i) (hidden) relationships in the routing information, in the topology information as well as between routing and topology information, ii) features/properties characterizing routing and topology information, and iii) classes in this information.

3) Resolve Routing Path function

The Resolve Routing Path function performs (hence it is also referred to as Produce Routing Path function) on the discovered (and possibly pre-processed) information to actually obtain the routing path by means of:

a) Computation function

This function is applied to (structured) routing information units and/or (structured) topology information units and that produces routing paths from which routing table entries can be derived. Computation can be seen as the operation of finding the routing path that minimizes/maximizes a (multi-)constrained (multi-)objective function. The computation function can be further sub-divided as follows:

- Compute Global-Full function: computation is based on global knowledge of the graph; upon routing information update the full routing table is recomputed.
- Compute Global-Incremental function: computation is based on global knowledge of the graph; upon routing information update only the affected routing table entries are recomputed.
- Compute Local function: computation is based on local knowledge (neighbors related information) possibly complemented by a partial knowledge of remote/non-local parts of the network graph properties.
- Compute Sequential function: computation is based on hop-by-hop/serial information propagated by local neighbors along either a given spatial trajectory.

b) Selection function

The selection function performs either by enforcing selection rules by applying filters and/or by multi-criteria decision on a set of routing information units, typically, routing paths with associated attributes. By means of this processing, a limited number of routing paths is selected from which routing table entries can be derived.

- Select/Filter Paths per Destination function: routing paths are selected per destination, e.g., path-vector based routing protocols.
- Select/Filter (logical) Ports per Destination function: (logical) ports are selected per destination, e.g., spanning-tree based routing protocols.

4) Produce Routing Table Entry function

The Produce Routing Table Entry function derives a route from the computed and/or selected routing paths and generates the corresponding routing table entry from the (selected) route, together with the creation of a new entry or the update of an existing routing table entry in the Routing Table (RT). Each forwarding table entry is then subsequently derived from a sub-set of one or more routing table entries.

5) Associated functions

The set of associated functions include i) all FCAPS functions associated to the routing functionality including system configuration, administration, etc., ii) the transfer of routing table entries: a mechanism allowing to export the routing table entries towards the forwarding engine component and iii) the trigger or poll for renewal/update control/routing behavior of a node based on external or internal events.

Moreover, "addressing and reachability" information can be derived by means of distribution/discovery function part of the routing functional area itself or by means of an associated resolution system dictionary (white boxes). Indeed, we assume that the routing functional area operates on locators (identifier assigned to nodes and endpoints that designate their location in an internetwork). Henceforth, an associated resolution function is defined that can operate either on reachability information only (e.g., Host Identity Protocol (HIP) where IP addresses function as pure locators; the applications use Host Identifiers to name peer hosts instead of using IP addresses) or on the topology/routing information itself (e.g., name-independent compact routing is an example of locator/label-based routing augmented with a identifier-to-locator function).

- *Identification function*: assigns identifiers to nodes. These names can be either topology-dependent (locators) or topology-independent (names); a locator can take the form of a label, a topology-dependent address or a coordinate.
- *Resolution function*: performs translation, conversion, or mapping from the name of the destination to its associated locator.
- *Location function*: the functionality allowing destinations to be located by means of the resolution function.

B. Routing Information Model

A generic routing information model elaborated by means of the E-R technique (depicted in Fig.2) comprises the following entities and relationships.

1) Entities

- Exchanged Information: information exchanged and communicated between routers part of the routing system. The exchanged topology information is structured in units, referred to as exchanged information units.
- Discovered Information: a crucial entity of the information model; this information is processed to compute and/or select routing paths.
  - Topology Information: includes pattern of interconnections of the various network components (e.g., node and links) which can be either physical or logical, i.e., the information related to local and remote interfaces (and their properties), incident links (and their properties), nodes adjacent to incident links (and their properties) as well as non-local environment information including remote links and (abstract) nodes. The discovered topology information is structured in units, referred to as topology information units. Topology information can be further sub-divided into Local (Neighbor) Information (entity denoting the information about the topology of the network is the immediate neighborhood of the local router) and Global (Network) Information (entity denoting the non-local information about the topology of the network for the corresponding router.
  - Routing Information: includes non-local distances (or information to derive these distances) and/or paths (segments) together with their attributes. The discovered routing information is structured in units, referred to as routing information units.
- Topology Information Base (TIB): structured entity comprising the collection of the locally stored TIB entries. A TIB entry is defined as an entity comprising a given topology information unit.
- Structured Information: results from the application of a combination of the following operations to the topology and/or routing information: composition (combination of topology and/or routing information so as to build more complex topology and/or routing information (called structured information), embedding/ mapping, and mining
- Routing Path: the path resulting from local routing computation algorithm or filtering. Routing path(s) attributes include navigation attribute(s), which contains the information about the direction to follow in order to reach the desired destination(s), a list of metric attributes, which characterize the different possible paths to follow. The navigation attribute is typically composed of a vector (e.g. path vector or distance vector) characterizing a set of possible destination, and a data structure composed of nodes (e.g., tree, list, etc.) containing enough information to reach the destination. The metric attributes can be of topological (e.g., number of nodes/routers, autonomous system), traffic engineering (e.g., bandwidth, delay, failure probability) or administrative (e.g., cost, color) nature.
- Route: entity derived from the routing path(s) that is subsequently used to route the traffic/packets. The set of route attributes include the metric values constituting the quantitative criteria on which the selection of particular routes is based. For example, this metric may be the number of hops (hop-count) to the destination. The destination designated by a locator or an identifier, may itself have associated attributes characterizing its nature. Moreover, depending on the routing protocol, additional attributes may complete this set.
- Selected Route: a subset of routes resulting from the application of route selection criteria (metric-selection, qualitative-selection, etc.). Their attributes are of the same type as those associated to Routes.
- Routing Information Base (RIB): the indexed collection of entries referred to as RIB entry, structured entity derived from the (selected) route; RIB entry attributes include the destination, the next-hop interface identifier together with other optional attributes including topological (e.g., routing path, hop-count), administrative (e.g. weight, cost) and traffic-engineering (e.g., bandwidth, delay) attributes.
  - RIB input: structured set of entities comprising all input (discovered or configured) routing information; in case of BGP for instance, this information includes routing paths and their associated attributes.
  - RIB output: structured set of entities comprising all locally computed and/or selected routing information (i.e., routing paths).
- Routing Table (RT): the indexed collection of entries individually referred to as RT entry. Each entry is defined as a structured entity comprising at least the next-hop node (or an indirection) to a particular destination and the associated metrics (that can be limited to a single metric value). The RT includes the entries produced according to the routing path computation/selection procedure(s) that are associated to a given routing protocol but also those

produced by the computation/selection procedures of other routing protocols.

2) Relationships

The following relationships are defined: Communicate/ Exchange, Derive, Generate, Filter, Owns/Includes, Produce, Structure, Select, and Transfer. The meaning of most relationships is self-explanatory. It should be noted that the meaning of some relationships overlap to some extent. For instance both Filter and Select relationships imply that the destination of the relationship is a carefully chosen part of the origin of the relationship.

VI. Application

To illustrate the utility of architectural modeling as well as the functional and information models specified in Section V, we document in this section their application to the BGP routing protocol and stochastic routing. Further explanations are provided in [27], which also details their application to unicast and multicast compact routing as well as geometric routing.

A. BGP Routing

Specified in RFC 4271 [3], the Border Gateway Protocol (BGP) relies on the path-vector routing algorithm. This routing protocol is used to exchange network reachability information between autonomous systems (AS). An AS is defined as "a set of routers under the control of a single technical administration entity or unit that presents a consistent picture of what destinations are reachable through it." Each AS, identified by its globally unique AS Number (ASN), comprises one or more border routers that connect to routers in neighboring AS, and possibly a number of internal BGP routers. The main function of BGP is to exchange network reachability information with peering (neighboring) BGP routers. Reachability information includes an AS path that lists the sequence of AS numbers traversed by the BGP route advertisement comprising reachability information from the originating AS. This information is used by BGP routers for constructing AS connectivity graph for this reachability so as to detect and avoid routing loops. BGP connections between routers belonging to neighboring AS are called eBGP (external BGP), while those between routers in the same AS are called iBGP (internal BGP). Note that adjacent AS may have more than one eBGP connection.

1) Theory of Operation

In BGP, a route is defined as a unit of information that pairs a set of destinations with the attributes of a path to those destinations. These routes are advertised between BGP routers in UPDATE messages. The set of destinations are systems whose IP addresses are contained in one IP address prefix that is carried in the Network Layer Reachability Information (NLRI) field of an UPDATE message. The actual path to this set of destinations is the information reported in the AS_PATH attribute field of the same UPDATE message. The AS_PATH attribute enumerates the sequence of AS numbers a route in the UPDATE message has traversed. As part of the set of mandatory attributes, we can mention the ORIGIN (generated by the BGP speaker that originates the associated routing information), the NEXT_HOP (defines the IP address of the router that should be used as the next hop to the destinations listed the UPDATE message), and the LOCAL_PREF (used by an iBGP speaker to inform its peers within the same AS of the advertising speaker's degree of preference for an advertised route). The MULTI_EXIT_DISC (MED) is an optional non-transitive attribute whose value may be used by a BGP speaker on external (inter-AS) links to discriminate among multiple exit or entry points to the same neighboring AS. In addition to the AS_PATH, the LOCAL_PREF and the MED attribute are also used in the BGP Route Selection process.

BGP routers (or speakers) advertise network reachability information about destinations by sending to their neighbors UPDATE messages containing a set of destination address prefix announcements and/or withdrawals together with the attributes associated to a path to these destinations.

- An announcement informs neighboring BGP routers of a path to a given destination. When a local BGP router propagates a route learned from the UPDATE message sent by one of its peering BGP routers, it modifies the route's AS_PATH attribute based on the location of the BGP router to which the UPDATE message containing that route will be sent.
- A withdrawal is an update indicating that a previously advertised destination is no longer reachable. Route withdrawals only contain the destination and implicitly tell the receiver to invalidate (or remove) the route previously announced by the sender. According to the above definition, if there is more than one path per destination, each path will be associated to a distinct route.

When a local BGP router propagates a route learned from the UPDATE message sent by one of its peering BGP routers, it modifies the route's AS_PATH attribute based on the location of the BGP router to which the UPDATE message containing that route will be sent. In contrast, route withdrawals only contain the destination and implicitly tell the receiver to invalidate (or remove) the route previously announced by the sender. A BGP router receives UPDATE messages from its BGP peering neighbors following a time varying interval bound by a minimum threshold. As detailed in RFC 4271 [3], there is a minimum amount of time (referred to as the Minimum Route Advertisement. Interval or MRAI) that must elapse between two UPDATE messages (for the same destination prefix) sent towards the same BGP router. Thus, a given BGP router receives one UPDATE message per MRAI time interval per neighbor (and sometimes per destination prefixes). The output (i.e., the class of the UPDATE message) of the learning process is used by the selection process of the local BGP router. This output is not distributed to the router's neighbors or other nodes in the system. However, the router's output (i.e., the BGP update messages that are forwarded) will influence the route selection of the BGP router's neighbor.

BGP routes are stored in Routing Information Bases (RIBs). At each BGP speaker, the RIB consists of three distinct parts: the Adj-RIB-In, the Loc-RIB, and the Adj-RIB-Out. The Adj-RIB-In contains unprocessed routing information that has been advertised to the local BGP speaker by its peers; the Loc-RIB contains the routes that have been selected by the local BGP speaker's decision process and the Adj-RIB-Out organizes the routes for advertisement to specific peers (by means of the local speaker's UPDATE messages). When a router receives a route advertisement, it first applies inbound filtering process (using some import policies) to the received routing information. If accepted, the route is stored in the Adj-RIB-In. The collection of routes received from all neighbors (external and internal) and stored in the Adj-RIB-In defines the set of candidate routes (for that destination). Subsequently, the BGP router invokes a route selection process - guided by locally defined policies - to select from this set a single best route for each destination. After this selection is performed, the selected best route is stored in the Loc-RIB and is subjected to some outbound filtering process and then announced to all the router's neighbors. Importantly, prior to being announced to an external neighbor, but not to an internal neighbor in the same AS, the AS path carried in the announcement is prepended with the ASN of the local AS.

2) *Functional Model*

Following the BGP routing protocol as specified in RFC 4271 [3] and outlined in Section VI.A.1, the BGP routing functionality can be hierarchically decomposed into three main functional blocks: i) the discovery (push) of routing information, i.e., BGP routes advertized by BGP peers, together with the optional inbound filtering of the received BGP routes, ii) the per-node selection of the "best" route by means of the so-called BGP route selection process, and iii) the push of the selected route together with optional outbound filtering of the selected BGP routes to downstream neighbors/ BGP speakers.

The BGP routing protocol makes thus use of the following functional blocks (see Fig.3): the Discover Routing Information function, the Produce Routing Path function, and the Produce Routing Table Entry function.

- Discover Routing Information function: enables the discovery of routing information which comprises the set of destination IP address prefix that is carried in the Network Layer Reachability Information (NLRI) field of an UPDATE message, the actual path to this set of destinations in the AS_PATH attribute field of the same UPDATE message together with mandatory and optional attributes associated to this path. Optionally, the incoming routing information is filtered by means of inbound or import filters before being stored in the Adj-RIB-In.

- Produce Routing Path function: performs by means of route selection function which applies either by enforcing selection rules on a set of (inbound filtered) routes stored in the Adj-RIB-In. By means of this selection process, a single route per destination is selected.

- Produce Routing Table Entry function: derives a route from the selected route and generates a Loc_RIB entry for this route or updates an existing entry in the Loc_RIB. Finally, the corresponding routing table entry is created or updated.

3) *Information Model*

In this section, we apply the generic information model detailed in Section.V.E to the BGP path-vector routing protocol outlined in Section VI.A.1. As depicted in Fig.3, where rectangles represent entities, diamonds relationships, and ovals attributes, this example shows the applicability of the proposed generic information model

4) *Observations*

Several observations can be drawn from these models:

- The exchange/discovery process is asymmetric: the RIB_In is actually decoupled from the Loc_RIB whereas the RIB_Out is driven by the selection/update rate of routing entries. The subsequent addition of a threshold to the routing update rate (i.e., the MRAI) at the sender-side is certainly a direct transposition of the "be liberal in what you accept and be conservative in what you send" design principle but in the meantime, the ratio RIB/FIB (function of the number of BGP peering sessions per BGP speaker) can easily reach an order of 10 (if not more since the number of BGP peering sessions is independent of the number of physical interfaces). Thus, routers have often to process an order to 10M routing entries to derive about 450k active routing table entries. Remember that the BGP update process "pushes" routing updates to neighbors. This mechanism defines probably the most basic technique for routing (data)base synchronization but its simplicity may actually be the root cause of the memory size scaling and adaptation cost observed nowadays. This observation leads to possibly rethink the routing update distribution process and not (only) the route selection process.

  In turn, this revisited process can significantly mitigate the routing path exploration phenomena which delay routing state convergence.

- The BGP route selection being driven by a node-based decision process, little flexibility is left to update neighbors on a per interface-basis beside application of outbound filters. This design is certainly desirable for inbound BGP speakers (with respect to the flow of routing updates) peering with BGP routers belonging to the same AS but less robust for outbound routers peering with different AS's.

- The nature of the routing update information (and its distribution process) is prone to induce path exploration; the question that stems though is why selecting a route subject to path exploration at first place. The answer is essentially because i) routing update information processing does not differentiate between updates with respect to their root cause, their identification (origin), etc. during the route selection process, and ii) the route selection process itself performs solely by applying network-wide criteria on the spatial properties of the AS-Path attributes (carried in routing updates) that are assumed to be immutable when processed. Thus, in addition to the routing update process itself, the information it distributes would have to be extended to incorporate temporal and infer causal properties.

- The BGP route selection process performs "on-path" regarding the flow of routing updates. This design choice seriously compromises the possibility for introducing any simple routing information verification mechanism crucial for security reasons (as a routing path and its associated routing update flow are congruent). Such mechanism aims at enabling the receiving BGP router to verify that the originating AS is authorized to advertise an address prefix by the holder of that prefix, whether the originating AS is accurately identified by the originating AS Number (ASN) in the advertisement, and the validity of both the address prefix and the ASN.

The proposed architectural modeling approach does not directly answer the question whether redesigning a new path-vector protocol or modifying BGP would be simpler to realize or not. Nevertheless, it provides the answer to the following questions: where to perform changes in the BGP functional and informational model design and how to design an alternative path-vector routing protocol; enabling in turn to compare and analyze both alternatives. Finally, the application of this architectural modeling approach to new routing schemes as documented in [27] provides a strong basis for comparison and analysis between them and BGP.

VII. Conclusion

In order to provide the architectural baseline of the Internet routing system, this document proposes to follow a systematic modeling approach relying on the specification of a functional and an information generic model. Indeed, past experience in designing routing protocols shows that without well defined system architecture, extending, adding or removing routing functionality leads to further complexity without actually meeting the intended scalability, adaptivity and performance objectives. More generally, finding the suitable tradeoff between short-term flexibility/adaptivity, long-term evolutivity, and performance is critical to ensure longevity of the routing system architecture. In this context, a top-down holistic approach for the design of the routing scheme (and its protocol components) complements the bottom-up algorithmic design approach.

The proposed generic functional and information models structure and, in turn, increase the cohesion between the different components that the specification of a routing scheme requires; this, by taking into account the challenges of distribution (resulting from the need to scale to large topologies partitioned into different units of operation) and adaptivity (resulting from dynamicity of the topology). We have shown the applicability of these models to the BGP path-vector routing protocol, and as documented in [27], their applicability to geometric routing, unicast and multicast compact routing but also stochastic routing. It is also interesting to observe that the first levels of the functional specification do require relatively limited per-routing scheme customization. More precisely, the customization mostly relates to the exchange mode of routing and topology information (and associated control), the pre-processing of exchanged/discovered information and obviously the specifics of the routing path computation algorithm itself. Further, and corroborating this observation, distributed and adaptive routing schemes shows the importance of the discovery function (and related information exchanges) which leads to reconsider the objects these routing schemes use to build the data structures on which routing path computation/selection performs. At this point, the rationale and the purpose of the information model become clearer, i.e., this model aims at specifying the information units, their properties and attributes so as to enable distributed computation and adaptivity of the routing decisions. Note also that in distributed systems like the Internet, the routing decisions are locally performed by each (abstract) node independently of the others using the exchanged information (i.e., discovered information) but individual nodes decision affects other router's decision. It is therefore fundamental to capture these interactions as part of the functional model up to the level appropriate for further routing system and protocol engineering.

The proposed approach enables also to thoroughly identify which routing (sub-)functions are currently under-specified or mis-specified but also which routing (sub-)functions can be replaced, added or even removed from their specification as documented in existing scientific literature. Comparison between the different routing schemes (and associated operations) is also facilitated as functional modeling offers at the same time a detailed functional analysis grid. Complementarily, the information model provides the mean to perform a detailed information analysis. In order to reach our ultimate goal of routing model specification, the present work will be further progressed by the specification of a procedural model (formal description of procedures) and a data model (formal description of data structures and their relationships together with data operators applied to these structures).

Acknowledgment

This research work is conducted and funded by the European Commission through the EULER project (Grant No.258307) part of the Future Internet Research and Experimentation (FIRE) objective of the Seventh Framework Programme (FP7).

References

[1] D.Clark, et al., Towards the Future Internet Architecture, Internet Engineering Task Force (IETF), RFC 1287, Dec.1991.

[2] BGP Report. Available at http://bgp.potaroo.net/index-bgp.html

[3] Y.Rakhter, T.Li, and S.Hares, Ed., A Border Gateway Protocol 4 (BGP-4), Internet Engineering Task Force (IETF), RFC 4271, Jan.2006.

[4] C.Labovitz, A.Ahuja, A.Bose, and F.Jahanian, Delayed internet routing convergence. Proc. of ACM SIGCOMM'00, pp.175-187, Stockholm, Sweden, Sep.2000.

[5] C.Labovitz, A.Ahuja, A.Bose, and F.Jahanian, Delayed Internet Routing Convergence, IEEE/ACM Transactions on Networking, Vol.9(3):293-306, 2001.

[6] C.Labovitz, R.Wattenhofer, S.Venkatachary, and A.Ahuja, The impact of Internet policy and topology on delayed routing convergence, Proc. of 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'01), pp.537-546, Anchorage (AK), USA, Apr.2001

[7] C.Labovitz, R.Malan, and F.Jahanian, Origins of Internet Routing Instability, Proc. of 18th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'99), pp.218-226, New York (NJ), USA, Mar.1999.

[8] T.Griffin, F.B.Shepherd, and G.Wilfong, The Stable Paths Problem and Interdomain Routing, IEEE/ACM Transactions on Networking, 10(1):232-243, 2002.

[9] D.Papadimitriou, A.Cabellos, and F.Coras, Path-vector Routing Stability Analysis, Proc.13th Workshop on MAthematical Performance Modeling and Analysis, ACM SIGMETRICS, San Jose (CA), USA, Jun.2011.

[10] C.Villamizar, et al, BGP Route Flap Damping, Internet Engineering Task Force (IETF), RFC 2439, Nov.1998.

[11] Z.M.Mao, R.Govindan, G.Varghese, and R.Katz, Route Flap Damping Exacerbates Internet Routing Convergence, Proc of ACM SIGCOMM'02, Pittsburgh (PA), USA, Aug.2002.

[12] J.Chandrashekar, Z.Duan, Z.-L.Zhang, and J.Krasky, Limiting path exploration in BGP, Proc. IEEE INFOCOM 2005, Miami, Florida, Mar.2005.

[13] D.Pei, M.Azuma, D.Massey, and L.Zhang, BGP-RCN: improving BGP convergence through root cause notification, Computer Networks, ISDN Syst. vol. 48, no.2, pp.175-194, Jun.2005.

[14] D.Walton, et al., Advertisement of Multiple Paths in BGP, Work in progress, draft-walton-bgp-add-paths, Jan.2009.

[15] D.Peleg and E.Upfall, A Trade-off between Space and Efficiency for Routing Tables, Journal of the ACM, vol.36, no.3, pp.510–530, 1989.

[16] I.Abraham, C.Gavoille, D.Malkhi, N.Nisan, and M.Thorup, Compact name-independent routing with minimum stretch, ACM Transactions on Algorithms (TALG), vol.4, no.3, art.37, Jun.2008.

[17] D.Krioukov, K.C.Claffy, K.R.Fall, and A.Brady, On Compact Routing for the Internet, ACM SIGCOMM Computer Communication Review (CCR), Vol. 37(3), 2007.

[18] M.Boguna, F.Papadopoulos, and D.Krioukov, Sustaining the Internet with Hyperbolic Mapping, Nature Communications, art.1, p.62, 2010.

[19] D.Krioukov, F.Papadopoulos, M.Boguna, and A.Vahdat, Efficient Navigation in Scale-Free Networks Embedded in Hyperbolic Metric Spaces, CAIDA, San Diego (CA), USA, May.2008.

[20] R.Kleinberg, Geometric routing using hyperbolic space, In Proceedings of the 26th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM'07), pp.1902-1909, Anchorage (AK), USA, May.2007.

[21] D.E.Perry, and A.L.Wolf, Foundations for the Study of Software Architecture, ACM SIGSOFT Software Engineering Notes, Vol.17, No.4, Oct.1992.

[22] D.Garlan and D.E.Perry, Editorial to the IEEE Transactions on Software Engineering, Apr.1995.

[23] G.Booch, Presentation at the Software Developers Conference, 1999.

[24] Techniques of Functional Analysis, pp.139-140. NASA, Jun.1995.

[25] P.Chen, "The Entity-Relationship Model--Toward a Unified View of Data". In: ACM Transactions on Database Systems 1/1/1976 ACM-Press ISSN 0362-5915, S. pp.9–36.

[26] D.Buede, Engineering Design of Systems - Models and Methods, John Wiley & Sons, 2000.

[27] EULER FP7 Project, Routing system architecture, Technical report, Available at: https://www-sop.inria.fr/mascotte/EULER/wiki/ pmwiki.php/ Main/Deliverables.

Annex 2: Paper "Stability metrics and criteria for path-vector routing"

Stability metrics and criteria for path-vector routing

Dimitri Papadimitriou
Alcatel-Lucent Bell Labs
Antwerp, Belgium
E-mail: dimitri.papadimitriou@alcatel-lucent.com

Albert Cabellos-Aparicio, Florin Coras
Technical University of Barcelona
Barcelona, Spain
E-mail: {acabello,fcoras}@ac.upc.edu

***Abstract*—Since so far, most studies on path-vector routing stability have been conducted by means of ad-hoc analysis of Border Gateway Protocol (BGP) data traces. None of them consider the specification of an analytic method including the use of stability metrics for the systematic analysis of BGP traces and associated meta-processing for determining the local state of the routing system. In this paper, we define a set of stability metrics that characterize the local stability properties of path-vector routing such as BGP. By means of these metrics, we derive a stability decision criterion that can be applied during the BGP route selection process. Results obtained using real BGP datasets show that 90% of the routes are not affected by a path length increase when selected based on this criterion. Moreover, among the remaining 10%, a significant fraction of the routes is covered by a path length increase of one-hop. These results corroborate the assumption that enforcing stability would not come at the detriment of increasing the stretch of the routing paths.**

Keywords-component; path-vector, routing, stability, metrics

I. Introduction

Prominent research efforts to understand Border Gateway Protocol (BGP) instability led to classify them as policy- or protocol-induced to account for the distinction between protocol operations and the inherent behavior of the underlying path-vector routing algorithm.

Policy-induced instabilities: solving the routing stability problem consistently with planned BGP routing policy requires to prevent and/or to eliminate conflicting policy interactions, in particular those leading to unintended unstable routing states. Griffin et al.'s seminal work [1] modeled BGP as a distributed algorithm for solving the Stable Paths Problem, and derived a general sufficient condition for BGP stability, known as "No Dispute Wheel". This sufficient condition guarantees the existence of a stable solution to which BGP always converges. Informally, this sufficient condition allows nodes to have more expressive and realistic preferences than always preferring shorter routes to longer ones. The game theoretic approach introduced in [2] relies on the best-reply BGP dynamics: a convergence game model in which each Autonomous System (AS) is instructed to continuously execute the following actions: i) receive update messages from BGP peering nodes announcing their routes to the destination, ii) choose a single peering node whose route is most preferred to send traffic to, iii) announce the new route to peering nodes. However, as proved in [2], best-reply BGP dynamics is not incentive-compatible even if No Dispute Wheel condition holds: even if all but one AS are following the BGP rules, the remaining AS may not have the incentive to follow them. Interestingly, as demonstrated in [2], incentive compatibility of best-reply BGP dynamics requires combining an additional global condition (Route Verification) together with the "No Dispute Wheels" to guarantee stability. Consequently, all known conditions for global stability define sufficient but not necessary conditions (checking them is an NP-hard problem and enforcing them requires a global deployment of an additional mechanism); on the other hand, local instability effects have yet to be characterized.

Protocol-induced instabilities: BGP, the inter-AS path-vector routing protocol of the Internet, is prone to Path Exploration, phenomenon characterizing any routing protocol that relies on the path-vector algorithm. Indeed, BGP routers may announce as valid, routes that are affected by a topological change and that will be withdrawn shortly after subsequent routing updates. This phenomenon is the main reason for the large number of routing updates received by BGP routers which exacerbate the inter-domain routing system instability and processing overhead [3]. Both result in delaying BGP convergence time upon topology change/failure [4]. Several mitigation mechanisms exist to partially limit the effects of path exploration; however, none actually eliminate them. Hence, BGP is intrinsically subject to instability.

As reported in RFC 4984, outcome of the Routing and Addressing Workshop held by the Internet Architecture Board (IAB) in 2006, BGP stability remains a key criterion to be met by the Internet routing system. It is also important to underline that the dynamics of the Internet routing system determines the resource consumption of routing engines, in particular, in terms of memory and CPU. System resource consumption depends on the size of the routing state space but also on the number of BGP peering relationships between routers. Indeed, the increasing dynamics of the exchanges of routing information updates between all BGP peerings, increases the memory and CPU requirements for the operations of the routing protocol. The objectives for investigating path-vector routing stability are to 1) Develop a method to systematically process and interpret the data part of BGP routing information bases in order to identify and characterize occurrences of BGP routing system instability from its routing paths properties; 2) Define a consistent set of stability metrics and related processing methods to better understand the BGP routing system's stability; 3) Exploit some of these metrics as route selection criteria. Overall, the proposed analytical method aims to bring rigor and consistency to the study of the stability properties of routing paths as locally experienced by routers.

This paper is structured as follows. Section II provides an overview on prior work concerning the BGP routing system stability. In Section III, we define the proposed routing stability metrics and detail the corresponding computational procedures. In this section, we also derive a stability decision criterion that can be applied during the BGP route selection process. In Section IV, we document the measurement methodology and the BGP datasets considered to evaluate the applicability of these metrics. Section V reports on the measurement results and analysis the applicability of the proposed stability-based criterion using real BGP datasets. Finally, Section VI draws conclusion from this study and outlines possible future work.

II. Prior Work

Beside the references cited in Section I, there have been numerous studies of BGP dynamics properties over the years. Work began in the early 1990s on an enhancement to the BGP called Route Flap Damping (RFD). The purpose of RFD was to prevent or limit sustained route oscillations that could potentially put an undue processing load on BGP. At that time there was a belief that the predominate cause of route oscillation was due to BGP routing sessions going up and down because they were being carried on circuits that were themselves persistently going up and down [3]. This would result in a constant stream of route updates from the affected BGP sessions that could propagate through the entire network due to the network's flat addressing architecture. The first version of the RFD algorithm specification appeared in October 1993, updates and revisions lead to the publication of RFC 2439 in November 1998.

Mao et al. [5] published in August 2002 a paper that discussed how the use of RFD, as specified in RFC 2439, can significantly slowdown the convergence times of relatively stable routing entries. This abnormal behavior arises during route withdrawal from the interaction of RFD with "BGP path exploration'' (in which in response to path failures or routing policy changes, some BGP routers may try a sequence of transient alternate paths before selecting a new path or declaring the corresponding destination unreachable). Bush et al. [6] summarized the findings of Mao et al. [5] and presented some observational data to illustrate the phenomena. The overall conclusion of this work was to avoid using RFD so that the overall ability of the network to re-converge after an episode of "BGP path exploration" was not needlessly slowed.

More recently solutions such as the enhanced path vector routing protocol EPIC [7] propose to add a forward edge sequence numbers mechanism to annotate the AS paths with additional "path dependency'' information. This information is combined with an enhanced path vector algorithm to limit path exploration and to reduce convergence time in case of failure. EPIC shows significant reduction of convergence time and the number of messages in the fail-down scenario (a part of the network is disconnected from the rest of the network) but only a modest improvement in the fail-over scenario (edges failures without isolation). The main drawback of EPIC is the large amount of extra information stored at the nodes and the increase of the size of messages. Another solution, BGP with Root Cause Notification (RCN) [8] proposes to reduce the BGP convergence delay by announcing the root cause of a link failure location. This solution also offers a significant reduction of the convergence time in the fail-down scenario. However, the convergence time improvement achieved with RCN is modest on the Internet topology compared to legacy BGP (in the fail-over scenario). More advanced techniques such as the recently introduced Path Exploration Damping (PED) [9] augments BGP for selectively damping the propagation of path exploration updates. PED selectively delays and suppresses the propagation of BGP updates that either lengthen an existing AS Path or vary an existing AS-path without shortening its length.

All these approaches try to mitigate the effects and/or to accelerate the convergence of the BGP routing entries after occurrence of a perturbation event, but none of them ask the fundamental question why selecting a route subject to path exploration at first place. The answer is essentially because these mechanisms rely on the network-wide quality criteria that are primarily based on the spatial properties of the AS-path.

III. Routing Stability and Metrics

A. *Preliminaries*

The autonomous system (AS) topology of the routing system is described as a graph G = (V,E), where each vertex (or node) u ∈ V, |V| = n, represents an AS, and each edge e ∈ E, |E| = m, represents a link between an AS pair denoted (u,v), where u, v ∈ V. At each node u ∈ V, a route r per destination d (d ∈ D) is selected and stored as an entry in the local routing table (RT) whose total number of entries is denoted by N, i.e., |RT| = N. At node u, a route r_i to destination d at time t is defined by $r_i(t) = \{d, (v_k{=}u, v_{k-1},\ldots,v_0{=}v), A\}$ with $k > 0 \mid \forall j, k \geq j > 0, \{v_j, v_{j-1}\} \in E$ and $i \in [1,N]$, where $(v_k{=}u, v_{k-1},\ldots,v_0{=}v)$ represents the AS-path, v_{k-1} the next hop of v along the AS-path from node u to v, and A its attribute set. Let $P_{(u,v),d}$ denote the set of paths from node u to v towards destination d where each path p(u,v) is of the form $\{(v_k{=}u, v_{k-1},\ldots,v_0{=}v), A\}$. A routing information update leads to a change of the AS-path $(v_k, v_{k-1},\ldots,v_0)$ or an element of its attribute set A. Next, a withdrawal is denoted by an empty AS-path (ε) and A = ∅: {d,ε,∅}. According to the above definition, if there is more than one AS-path per destination d, they will be considered as multiple distinct routes.

B. *Routing Stability*

The stability of a routing system is characterized by its response (in terms of processing of routing information) to inputs of finite amplitude. Routing system inputs may be classified as i) internal system events such as changes in the routing protocol configuration or ii) external events such as those resulting from topological changes. Both types of events lead to the exchange of routing information updates (or simply routing updates) that may result in routing states changes. Indeed, BGP and in general any path-vector routing, does not differentiate routing updates with respect to their root cause, their identification (origin), etc. during its selection process.

Definition 1: Let $r_i(t)$ represent the route r_i at some time t as stored in the routing table (RT). At time t+1, $r_i(t+1) = r_i(t) \oplus \Delta r_i(t+1)$, where $\Delta r_i(t+1)$ accounts for all changes experienced by the route r_i from time t to t+1.

This research work is conducted and funded by the European Commission through the EULER project (Grant No.258307) part of the Future Internet Research and Experimentation (FIRE) objective of the Seventh Framework Programme (FP7).

Definition 2: Let RT(t) represent the routing table at some time t. At time t+1, RT(t+1) = RT(t) ⊕ ΔRT(t+1) where, RT(t) is the set of routes that experience no change between time t and t+1, and ΔRT(t+1) accounts for all route changes (additions, deletions, and changes to previously existing routes) between time t and t+1.

The magnitude of the output of a stable routing system should be small whenever the input is small. That is, a single routing update shall not result in output amplification. Equivalently, a stable system's output will always decrease to zero whenever the input events stop. A routing system, which remains in an unending condition of transition from one state to another when disturbed by an external or internal event, is considered to be unstable. In this context, provide means for measuring the magnitude of the output is the main purpose of the metric referred to as "stability of the selected route". For this purpose, we define the criteria for qualifying the effects of a perturbation on the local routing table entries so as to locally characterize the stability properties of the routing system. More precisely, let |ΔRT(t+1)| be the magnitude of the change to the routing table (RT) between time $t = t_0 + k$ to $t + 1 = t_0 + (k + 1)$, where t_0 is the starting time of the measurement sequence, and k the integer that determines the number of Minimum Routing Advertisement Interval (MRAI) that have elapsed since the starting time of the measurement sequence. The MRAI determines the minimum amount of time that must elapse between an advertisement and/or withdrawal of routes to a particular destination by a BGP speaker to a peer. The MRAI does not limit the rate of the route selection process but only the rate of route advertisements. Hence, using the MRAI as time unit ensures to record at most one routing update per destination (per BGP peer) per sampling period. Using these definitions, we distinguish three different equilibrium states for the routing table:

Definition 3: when disturbed by an external and/or internal event, a RT is considered to be *stable* if: |ΔRT(t+1)| ≤ α, t → ∞, where α > 0 is small. If this condition is met, the routing system (as locally observed) returns to its initial equilibrium state, and is considered to be (asymptotically) stable.

Definition 4: when disturbed by an external and/or internal event, a RT is considered to be *marginally stable* if: α < |ΔRT(t+1)| ≤ β, t → ∞, where β > 0 is small, α < β. If this condition is met, the routing system (as locally observed) transitions to a new equilibrium state, and is considered to be marginally stable.

Definition 5: when disturbed by an external and/or internal event, a RT is considered to be *unstable* if: |ΔRT(t+1)| > β, t → ∞. If this condition is met, the routing system (as locally observed) remains in an unending condition of transition from one state to another, and is considered to be locally unstable

The actual values of the parameters α and β depend on several factors. Among them, the MRAI value and the integer k that determines the number of MRAI that have elapsed since the beginning of the observation sequence. Other factors influencing these parameters are explained in Section III.C.

Note that a similar reasoning to the one applied for the Loc_RIB stability (that corresponds to the BGP routing table) can be applied to the Adj_RIB_In, which stores the incoming routes from neighbors. It is also interesting to measure the instability induced by the BGP selection process itself.

C. *Stability Metrics*

To measure the degree of stability of the Loc_RIB, and the Adj_RIB_In, the following stability metrics are introduced.

The *stability* $\varphi_i(t)$ of the selected route $r_i(t)$ characterizes the stability of the route $r_i(t)$ ($i \in [1,|D|]$) stored at time t in the Loc_RIB (|Loc_RIB| = N). The value $\varphi_i(t)$ quantifies the magnitude of the change(s) experienced by the route r_i from time $t = t_0+k$ to time $t+1 = t_0+(k+1)$, where t_0 is the starting time of the measurement sequence (time units are counted by default in terms of MRAI), and the integer k accounts for the number of MRAI times that have elapsed since the starting time of the measurement sequence. This metric quantifies thus the magnitude of the change(s) experienced by the route r_i with a periodicity determined by the MRAI time. This metric can be computed by using the procedure described in Fig.1. Upon creation of a new routing table entry associated to the route r_i, the value $\varphi_i(t)$ is initialized together with the parameters α and β (see Section III.B). These parameters can be derived from this procedure on a per individual route basis.

```
/* Initialization when route $r_i$ is created */
$\varphi_i(t) \leftarrow 0$
$\alpha_{min,i} = \alpha_i \leftarrow 0$
$\beta_{max,i} = \beta_i \leftarrow 0$

/* Measurement during k * MRAI time units */
While k > 0
if $\Delta r_i(t+1) \neq 0$
   /* $r_i$ experiences an AS-path change
      or $r_i$ experiences an attribute change */
then $\varphi_i(t+1) \leftarrow \varphi_i(t) + 1$
     $\beta_i \leftarrow \varphi_i(t+1)$
     if $\beta_{max,i} < \beta_i$ then $\beta_{max,i} \leftarrow \beta_i$
     end if
else /* $r_i$ experiences no change: $\Delta r_i(t+1)=0$ */
     if $\varphi_i(t) > 0$
     then $\varphi_i(t+1) \leftarrow \varphi_i(t) - 1$
          $\alpha_i \leftarrow \varphi_i(t+1)$
          if $\alpha_i > \alpha_{min,i}$ then $\alpha_{min,i} \leftarrow \alpha_i$
          end if
     else $\varphi_i(t+1) \leftarrow 0$
     end if
end if
k ← k - 1
end k loop
```

Figure 1. Stability of individual routes

As described in Fig.2, the computation of the stability metric for an entire routing table can then be derived form the stability of its individual routes. Let $|\Delta r_i(t+1)|$ denote the magnitude of change in terms of stability as experienced by a single route r_i from time t to t+1. The set of values $|\Delta r_i(t+1)|$, i

$\in$ [1,N], are then used to compute the value $|\Delta RT(t+1)|$ defined as the magnitude of change in terms of stability for the entire routing table from time t to t+1. Moreover, $|\Delta RT(t+1)|$ can be normalized so that $0 \leq |\Delta RT(t+1)| \leq 1$, where 0 implies perfect stability, and 1 indicates complete instability.

```
For i=1 to N
   /* N = total number of routes in RT(t+1) */
   if $r_i(t+1)$ is a new route
      $\vee$ [$\varphi_i(t) = 0 \wedge \varphi_i(t+1) = 0$]
   then $|\Delta r_i(t+1)| \leftarrow 0$
   else if $\varphi_i(t+1) > \varphi_i(t)$
        then $|\Delta r_i(t+1)| \leftarrow [\varphi_i(t)+1]/[\varphi_i(t+1)+1]$
        else if $\varphi_i(t+1) \leq \varphi_i(t)$
             then $|\Delta r_i(t+1)| \leftarrow \varphi_i(t+1)/\varphi_i(t)$
             end if
        end if
   end if
end i loop

$\mu = |\Delta RT(t+1)| \leftarrow \Sigma_i |\Delta r_i(t+1)| / N$
$\sigma^2 \leftarrow \Sigma_i (\Delta r_i(t+1) - |\Delta RT(t+1)|)^2 / N$
```

Figure 2. Stability metric computation for a set of routing entries

The *most stable route* in the Adj_RIB_In (|Adj_RIB_In| = M) quantifies the relative stability between incoming routes to the same destination d as learned from all upstream BGP peers (i.e., downstream from the point of view of the AS-path towards destination d) and the one amongst them determined as the most stable at time t. For this purpose, let $W_u \subset V$ denote the set of node's u BGP peers, $|W_u| = W \leq M$, and w one of its elements such that $(u,w) \in E$. Let $\varphi_{i,j}(t)$ denote the stability of the route $r_i(t)$ to destination d as received by the peering router j ($j \in [1,W]$) at time t. At node u, $r'_{i,stable}(t)=\min\{\varphi_{i,j}(t), \forall j \in [1,W] \mid \{(v_k=u,v_{k-1}=w,\ldots,v_0=v),A\} \in P_{(u,v),d}, \forall w \in W_u\}$ defines –independently of the BGP route selection rules– the selectable route that is the most stable for destination d at time t. Next, we define $\Delta\varphi_i$ as the relative measure stability $\varphi_{i,j}$ of the route r_i at time t+1 for destination d with respect to the stability $\varphi_{i,stable}$ of the most stable route $r'_{i,stable}$ at time t for the same destination d.

```
For i=1 to N
    /* |dest. in Adj_RIB_In| = |Loc_RIB| */
    For j=1 to $|W_u|$
        /* number of peers for i^th dest. */
        $\Delta\varphi_{i,j}(t+1) \leftarrow [\varphi_{i,j}(t+1)+1]/[\varphi_{i,stable}(t)+1]$
    end j loop
    $\Delta\Phi_i(t+1) \leftarrow \Sigma_j \Delta\varphi_{i,j}(t+1)/|W_u|$
end i loop

$\mu = \Delta\Phi(t+1) \leftarrow \Sigma_i \Delta\Phi_i(t+1) / N$
$\sigma^2 \leftarrow \Sigma_i (\Delta\Phi_i(t+1) - \Delta\Phi(t+1))^2 / N$
```

Figure 3. Most stable route

The *best selectable route* from the Adj_RIB_In quantifies the relative stability between incoming routes to the same destination d as learned from all upstream peers and the one amongst them selected by BGP at time t as the best route (thus, following BGP route selection rules). As described in Fig.4, the computational procedure is similar to the one depicted in Fig.3, if one replaces $\varphi_{i,stable}$ by $\varphi_{i,selected}$ during the computation of the $\Delta\varphi_{i,j}$ terms.

```
For i=1 to N
    /* |dest. in Adj_RIB_In| = |Loc_RIB| */
    For j=1 to $|W_u|$
        /* number of peers for i^th dest. */
        $\Delta\varphi_{i,j}(t+1) \leftarrow [\varphi_{i,j}(t+1)+1]/[\varphi_{i,selected}(t)+1]$
    end j loop
    $\Delta\Phi_i(t+1) \leftarrow \Sigma_j \Delta\varphi_{i,j}(t+1)/|W_u|$
end i loop

$\mu = \Delta\Phi(t+1) \leftarrow \Sigma_i \Delta\Phi_i(t+1) / N$
$\sigma^2 \leftarrow \Sigma_i (\Delta\Phi_i(t+1) - \Delta\Phi(t+1))^2 / N$
```

Figure 4. Best selectable route

The *differential stability* between the most stable route in the Adj_RIB_In and the selected route stored in the Loc_RIB for the same destination d characterizes the stability of the currently selected routes for a given destination d against most stable routes as learned from upstream neighbors. This metric provides a measure of the stability of the learned routes compared to the stability of the currently selected route. A variant of this metric, denoted $\delta\varphi_i(t)$, $i \in [1,|D|]$, characterizes the stability of the newly selected path p*(u,v) at time t for destination d against the stability of the path p(u,v) that is stored as time t in the Loc_RIB for destination d and that would be replaced at time t+1 by the path p*(u,v): $\delta\varphi_i(t) = \varphi_i(t) - \varphi_i^*(t)$. In turn, if the differential stability metric $\delta\varphi_i(t) > 0$, then the replacement of route $r_i(t)$ by the route $r_i^*(t)$ increases the stability of the route to destination d; otherwise, the safest decision is to keep the currently selected route $r_i(t)$ stored in the Loc_RIB.

Application of the differential stability metric $\delta\varphi_i$ during the BGP selection process would prevent replacement (in the Loc_RIB) of more stable routes by less stable ones but also enable selection of more stable routes than the currently selected routes. However, for this assumption to hold, we must also prove the consistency of the stability-based selection with the existing preferential-based route selection model that relies on a path ranking function (i.e., a non-negative, integer-value function λ_u, defined over $P_{(u,v),d}$, such that if $p_1(u,v)$ and $p_2(u,v) \in P_{(u,v),d}$ and $\lambda_u(p_1) < \lambda_u(p_2)$ then $p_2(u,v)$ is said to be preferred over $p_1(u,v)$). The route selection problem is consistent with the stability function $\delta\varphi(t)$, if $\forall$ $u \in V$ and $p_1(u,v)$ and $p_2(u,v) \in P_{(u,v),d}$ (1) if $\lambda_u(p_1) < \lambda_u(p_2)$ then $\delta\varphi(t) = \varphi_1(t) - \varphi_2(t) \geq 0$ and (2) $\lambda_u(p_1) = \lambda_u(p_2)$ then $\delta\varphi(t) = 0$. We show in [10] that if $p_1(u,v)$ and $p_2(u,v) \in P_{(u,v),d} \wedge p_2(u,v)$ is embedded in $p_1(u,v)$, then the route selection problem is consistent with the stability function $\delta\varphi$ and the route selection is not stretch increasing. By stretch decreasing, we mean here that the length $\rho_i^*(t)$ of the path p*(u,v) (measured in terms of number of AS hops in case of BGP route) associated at time t to the route r_i^* is smaller than the length $\rho_i(t)$ of path p(u,v) associated at time t to the route r_i: $\delta\rho_i(t) = \rho_i^*(t) - \rho_i(t) < 0$.

D. Stability Decision Criterion

The BGP selection process enhanced by the stability-based decision criteria, following the differential stability metric defined in Section III.C, would be driven by the following selection rules:

```
if $\delta\varphi_1(t) > 0$
then if $\delta\rho_i(t) \le 0$
     then select $r_1(t)$ per $\delta\varphi_1(t)$
     else if $\delta\rho_i(t) < \gamma$
             then select $r_1(t)$ per $\delta\varphi_1(t)$
             else select $r_1(t)$ per
                  default BGP selection rules

if $\delta\varphi_1(t) \le 0$
then select $r_1(t)$ per
     default BGP selection rules
```

In this selection process, the positive integer parameter γ is determined by the increase of the multiplicative stretch considered as acceptable. Hence, the actual problem becomes to find a mean to actually determine (or at least estimate) the acceptable stretch increase of the routing path that would result from the application of the stability-based decision criteria. Past experiments dedicated to the measure of the BGP AS-path length have shown that even if the average length of AS-paths is relatively stable (about 4 to 5), a significant fraction of AS-paths has a length up to 10 [11]. From this perspective, if we assume that a 10% increase of the multiplicative stretch would be acceptable (resulting multiplicative stretch would be equal to 1.1 instead of 1.0), then routes with an average AS-path length increase of 1 AS-hop would instead be selected. Note that this study does not evaluate the increase in memory consumption required to store the routes with longer AS-path attributes. Moreover, the application of the stability-based decision criterion prevents propagation of the routing updates churn resulting from the occurrence of a path exploration event when the following conditions are met i) the route corresponding to the next stable state is locally stored in the Adj_RIB_In and ii) this route corresponds to the most stable (next) route in the Adj_RIB_In. Indeed, if such event occurs, then the selection of a stable route becomes possible without delaying local convergence resulting from the exploration of all intermediate routing states (e.g., AS-paths of increasing length). Nevertheless, if the path exploration event also affects the route corresponding to the next state corresponding to the most stable next route, then selecting the AS-path that is the least topologically correlated[1] to the previous state provides the safest decision.

Importantly, the applicability of the stability-based decision criterion does not only depend on the point-value of the differential stability metric but also on its evolution over time. This means in practice that we have also to ensure that when the stability criteria are met at time t, and the corresponding selection rules are applied at time t, they also remain applicable at time t+1, and more generally at time t+Δt, where Δt >> 0. The reason stems as follows: at a given router once a route is selected at time t based on its stability properties, reverting unilaterally to the default BGP selection rules at time t+Δt can itself increase the instability induced by the concerned routes on its downstream routers. Here again, our stability metrics provide a suitable method to estimate the deviation over time and the robustness of the selection process. Indeed, it suffices to notice that (even if it is impossible to locally anticipate all occurrence of BGP instability events before they occur) these metrics enable to determine over time the candidate replacement routes that are more stable compared to the set of possible alternative routes that do not show the same stability properties. When such alternative route does not exist, the exchange process of BGP routing updates between the local router and its downstream neighbors (with respect to the direction of propagation of the routing updates) requires enhancement in order to enable a smooth transition between the route selection rules. This mechanism performs as follows: anticipatively once no candidate replacement route is available for the route currently selected based on the stability criteria, that route is advertized to downstream neighbors together with the route that would be selected based on the default BGP selection rules. This process enables each downstream router to tune its decision process based on its own selection rules for that route. Note that this process enables to advertize both routes, i.e., the one selected based on the stability criteria and the one selected based on the BGP default rules.

IV. Measurement Methodology and Data Set

We apply the metrics defined in Section III to the BGP updates provided by the Route-Views project [12]. The BGP dataset obtained from this project comprises archives containing BGP feeds from a set of worldwide distributed Linux PCs running Quagga/Zebra[2] [13]. Route Views is a project founded and sponsored by the University of Oregon which consists in a set of routers distributed across the world. The BGP routing information collected by these routers can be openly accessed by anyone, interested or involved in the field of Internet research. This information has led to various noticeable studies including those conducted in [3]. The Route-Views data records contain the BGP information a router receives from its neighboring BGP speakers. That is basically each neighbor route (with its route attributes) to each address prefix the neighbor has knowledge about. With this information, the monitored BGP router can find a route for each IP prefix it needs to send packets to.

Current Route-Views data record format does not provide the Loc_RIB information as stored locally by each router (that is, basically and for our interests, the information about which route BGP selects for each prefix). We must thus derive this information from the Adj-RIB-In table as provided by the Route-Views dataset. For this purpose, we infer the Loc-RIB table from the Adj-RIB-In tables by implementing a selection process based on the algorithm used in Quagga/Zebra routers, which is representative of the BGP selection process

[1] Two AS_paths are topologically correlated if they share at least one common edge, i.e., an AS adjacency.

[2] Quagga is a fork of GNU Zebra which was developed by Kunihiro Ishiguro.

commonly applied on Internet routers. A detailed description of the tool developed in C++ programming language to process the BGP datasets obtained and to derive the value of the metrics (defined in Section III), their associated statistics as well as their evolution over time is available in [14].

V. Measurement Results and Analysis

This section presents a set of experimental results obtained by applying the metrics and selection rules defined in Section III to BGP dataset obtained from the Route-Views project [12].

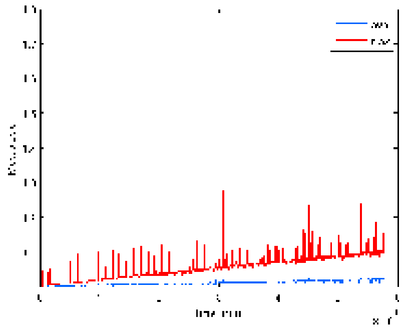


Figure 5. Most stable route metric measure

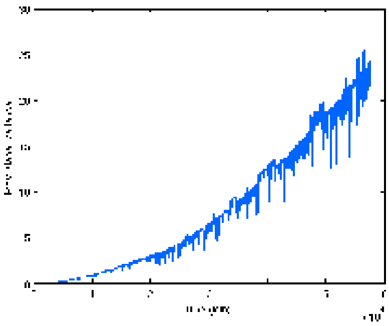


Figure 6. Cumulated variance over time for most stable route

Fig.5 shows that incoming routes stored in Adj_RIB_In have on average slowly decreasing stability compared to the most stable route (a value close to 1 indicates that incoming routes are nearly as stable as the most stable route). As a result, the plot has a small but positive slope. The average of the maximum metric value per destination d shows a positive but larger slope: the most unstable routes have a faster paced decreasing stability (and spiky pattern confirms their unstable behavior). Further, during the entire observation duration (40 days), a subset of routes continuously presented instabilities leading to a monotonic increase of the metric. It can be observed from Fig.5 that the BGP selected route has on average a better stability than the other routes out of which it is selected (a value close to 1 indicates that incoming routes are nearly as stable as the best selectable route). Comparison between Fig.5 and Fig.7 reveals though that local maxima for the selected route exhibits more spaced and less intensive variations than the most stable route (a lower metric value indicates a higher stability). One can also observe the same monotonously increasing trend of this metric for both the average and the maximum, due to routes with sustained instability.

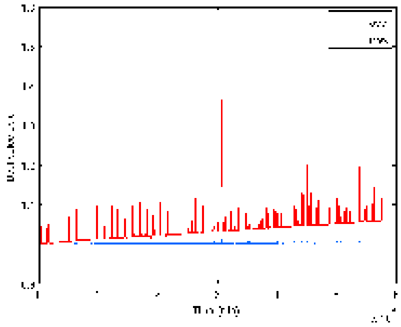


Figure 7. Best selected route metric measure

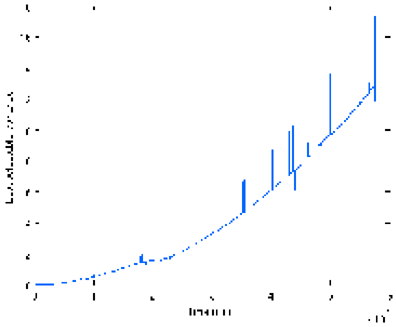


Figure 8. Cumulated variance over time for the best selectable route

Computation of the cumulated variance for the most stable route (Fig.6) shows an increase value over time from about 5 after 20 days and about 25 after 40 days; the slope is super-linear. Nevertheless, we can observe that after one day the variance remains relatively limited, leaving the possibility of selecting the most stable route without incurring significant stability deviation of the entire routing table. Local maxima in Fig.8 indicate large changes in local route stability, i.e., more routes than the average experience instabilities but BGP quickly converges to a new stable state since a part of the affected routes return to their initial state (thanks to the presence of more stable routes in the Adj_RIB_In, as indicated in Fig.6). Interestingly, Fig.8 shows also that the intensity of the instability increases over time indicating that more routes get affected by the change.

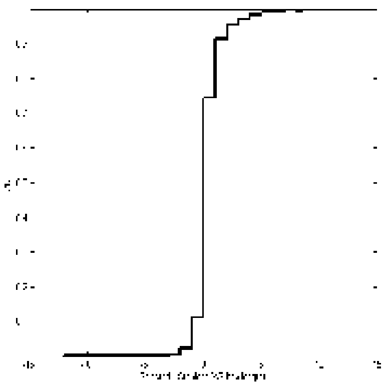


Figure 9. Number of routes vs diff. in AS-path length

Fig.9 shows the cumulated percentage of routes with respect to the AS-path length difference between the selected and the most stable route. A positive difference indicates that the replacement of the selected route (using the BGP path ranking function) by the most stable route would decrease the AS-path length compared to the selected route ($\delta p < 0$). A negative difference indicates that such replacement would increase the AS-path length ($\delta p > 0$). From this figure, we can deduce that such replacement would be advisable for about 90% of the selected routes since $\delta p \leq 0$. Moreover, for 25% percent of the routes, this replacement would also lead to an AS-path length decrease since for these routes $\delta p < 0$. Interestingly, only 10% of the routes would be affected by a length increase if they would be selected based on the stability criteria since for these routes $\delta p > 0$. Among this percentage of 10%, we can also observe from this figure that a significant fraction of the routes would be covered if an AS-path length increase of one-hop would be considered as acceptable (in average $\delta p \cong 1.15$). These observations corroborate the fact that the stability-based selection rule does not lead to a stretch increase for a significant fraction of the routes (90%). On the other hand, by admitting a stretch increase corresponding to one additional AS-hop in the AS-path, only a minor fraction of the routes (about 2%) would be penalized by a higher stretch increase of two AS-hops (and above for a fraction of routes << 1%). This observation can be seen as the experimental evidence that enforcing stability would not come at the detriment of increasing the stretch of the AS-paths.

VI. Conclusion

In this paper, we have defined several stability metrics to characterize the local effects of BGP policy- and protocol-induced instabilities on the routing tables. Our experimental results show that the proposed method enables to locally detect instability events that are affecting routing tables' entries, and deriving their impact on the local stability properties of the routing tables. We have also determined a differential stability-based decision criterion that can be taken into account as part of the BGP route selection process. A significant fraction of the routes (90%) selected by means of this process is not stretch increasing. Moreover, if one would admit an AS-path length increase of one AS-hop, only a minor fraction of the routes (about 2%) would be penalized by a higher stretch increase (two AS-hops and above).

Future work includes verifying the general trade-offs between stability-based route selection and the resulting stretch increase/decrease on the selected routing paths. Moreover, the relationship between local and global stability will be further elaborated to characterize the effects on the global stability of the routing system resulting from the selection of a route that is more stable locally. The idea here is to determine the necessary but sufficient conditions for preventing potential oscillations to occur (as the local action of selecting a more stable route shall not induce unwanted perturbation(s) on neighboring routing states). Finally, the model will also be extended to locally discriminate between protocol- and policy-induced instabilities.

References

[1] T.Griffin, F.B.Shepherd, and G.Wilfong, The Stable Paths Problem and Interdomain Routing, IEEE/ACM Transactions on Networking, 10(1):232-243, April 2002.

[2] H.Levin, M.Schapira, and A.Zohar, Interdomain routing and games, Proc. of ACM Symposium on Theory of Computing (STOC), 2008.

[3] G.Huston, Damping BGP, RIPE 55, Routing WG, October 2007.

[4] C.Labovitz, A.Ahuja, A.Bose, and F.Jahanian,, Delayed Internet Routing Convergence, IEEE/ACM Transactions on Networking, 9(3):293-306, June 2001.

[5] Z.M.Mao, R.Govindan, G.Varghese, and R.Katz, Route Flap Damping Exacerbates Internet Routing Convergence, Proc. of ACM SIGCOMM 2002, August 2002.

[6] R.Bush, T.Griffin, and Z.M.Mao, Route flap damping harmful?, NANOG-26, 28 October 2002.

[7] J.Chandrashekar, Z.Duan, Z.-L.Zhang, and J.Krasky, Limiting path exploration in BGP, Proc. of IEEE INFOCOM 2005, Miami, Florida, March 2005.

[8] D.Pei, M.Azuma, D.Massey, and L.Zhang, BGP-RCN: improving BGP convergence through root cause notification, Computer Networks, ISDN Syst. vol. 48, no.2, pp.175-194, June 2005.

[9] G.Huston, M.Rossi, and G.Armitage, A Technique for Reducing BGP Update Announcements through Path Exploration Damping, IEEE Journal on Selected Areas in Communications, vol.28, no.8, October 2010.

[10] D.Papadimitriou, A.Cabellos, and F.Coras, Path-vector Routing Stability Analysis, Proc.13th Workshop on MAthematical Performance Modeling and Analysis, ACM SIGMETRICS, San Jose (CA), USA, June 2011.

[11] B.Huffaker, M.Fomenkov, M.Plummer, D.Moore, and k.claffy, Distance Metrics in the Internet, IEEE International Telecommunications Symposium (ITS), Brazil, pp.200-202, September 2002.

[12] Univ. of Oregon. RouteViews. Available at: http://www.routeviews.org

[13] Quagga Routing Software Suite, GPL licensed, Available at: http://www.nongnu.org/quagga/

[14] EULER FP7 Project, Measurement-based Topology Modeling, Technical report, Deliverable D3.2, Available at: https://www-sop.inria.fr/mascotte/EULER/wiki/pmwiki.php/Main/Deliverables

Annex 3: Paper "Relationship between path-vector routing and forwarding path stability"

Relationship between path-vector routing and forwarding path stability

Dimitri Papadimitriou
Alcatel-Lucent
Antwerp, Belgium
E-mail: dimitri.papadimitriou@alcatel-lucent.com

Davide Careglio
Technical University of Barcelona
Barcelona, Spain
E-mail: careglio@ac.upc.edu

Fabien Tarissan
UPMC - LIP6
Paris, France
E-mail: fabien.tarissan@lip6.fr

***Abstract*—Analysis of real datasets to characterize the local stability properties of the Internet inter-domain routing paths suggests that extending the route selection criteria to account for such property would not increase the routing path length. Nevertheless, even if selecting a more stable routing path could be considered as valuable from a routing perspective, it does not necessarily imply that the associated forwarding path would be more stable. Hence, if the dynamics of the Internet routing and forwarding system show different properties, then one can not straightforwardly derive the one from the other. If this assumption is verified, then the relationship between the stability of the forwarding path followed by the traffic and the corresponding routing path as selected by the path-vector routing protocol requires further characterization. For this purpose, we locally relate, i.e., at the router level, the stability measurements carried on forwarding paths with the corresponding routing paths. Our results verify this assumption and show that, although the main cause of instability results from the forwarding plane, a second order effect relates forwarding and routing path instability events. This observation provides first indication that differential stability can safely be taken into account as part of the route selection process.**

Keywords-component; path-vector, routing, stability, metrics

I. Introduction

Following the Routing and Addressing Workshop held by the Internet Architecture Board (IAB) in 2006 [1], stability remains a key criterion to be met by the Internet routing system and its underlying Border Gateway Protocol (BGP). The prominent research efforts **Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.**[5] conducted over last fifteen years to understand BGP instabilities led to classify them as policy-induced or protocol-induced to account for the distinction between BGP protocol operations and the inherent behavior of the underlying path-vector routing algorithm. Following these studies, stability of the individual local routing states and associated routing path should remain (at least marginally) stable upon occurrence of perturbation resulting from i) the exploration of the routing state space due to the BGP path exploration phenomenon that is intrinsic to the shortest-path vector algorithm, and ii) the BGP routing policies interactions due to which among other can lead to "dispute wheels", i.e., non-deterministic unintended but unstable states. In this context, it is important to underline that the dynamics of the Internet routing system determines the resource consumption of local routing engines, in particular, in terms of memory and CPU. System resource consumption depends on the size of the routing state space but also on the number of BGP peering relationships between routers. Indeed, the increasing dynamics of the exchanges of routing information updates between all BGP peerings increases the memory and CPU requirements for the operations of the routing protocol.

The overall objectives for investigating path-vector routing stability are to 1) Develop a method to systematically process and interpret the data part of BGP routing information bases in order to detect, identify and characterize occurrences of BGP routing system instability from its routing paths properties; 2) Define a consistent set of stability metrics and related processing methods to better understand the BGP routing system's stability; 3) Exploit some of these metrics as possible route selection criteria. The method proposed in [5] aims to bring rigor and consistency when studying the stability properties of routing paths as locally experienced by routers. The experimental results reported show that this method enables to locally detect instability events that are affecting routing tables' entries, and deriving their impact on the local stability properties of the routing tables. From the metrics defined in [5], a differential stability-based decision criterion is derived that can be taken into account as part of the BGP route selection process [6]. Results show that a significant fraction of the routes (90%) selected by means of this process is not stretch increasing. Moreover, if one would admit an AS-Path length increase of one AS-hop, only a minor fraction of the routes (about 2%) would be penalized by a higher stretch increase (two AS-hops and above).

Nevertheless, even if selecting a more stable routing path could be considered as valuable from a routing level perspective, it does not necessarily imply that the corresponding forwarding path(s) would be itself more stable. In this work, our first objective consists thus in determining if the dynamics of the Internet routing and forwarding system (through the analysis of routing and forwarding path instability) show different properties. If this assumption is verified then as one can not straightforwardly derive the one from the other; our second objective becomes to investigate the relationship between the stability of the forwarding path followed by the traffic and the corresponding routing path as selected by the path-vector routing protocol. For this purpose, we locally relate at the router level, the stability measurements carried on forwarding paths with the corresponding routing paths following the method developed in [6].

The remainder of this paper is structured as follows. Section II provides an overview on prior work concerning the BGP routing system stability. In Section III, we review the proposed routing stability metrics and detail the corresponding computational procedures. We document in Section IV the measurement and processing methodology together with the real datasets onto which these metrics have been applied. Section V reports on the measurement results and analysis obtained. Finally, Section VI draws conclusion from this study and outlines possible future work.

II. Prior Work

Numerous studies on BGP dynamics properties have been conducted over last twenty years. Work began in the early 1990s on an enhancement to the BGP called Route Flap Damping (RFD). The purpose of RFD was to prevent or limit sustained route oscillations that could potentially put an undue processing load on BGP. At that time, the predominant cause of route oscillation was assumed to result from BGP sessions going up and down because established on circuits that were themselves persistently going up and down. This would lead to a constant stream of BGP update messages from the affected BGP sessions that could propagate through the entire network. The first version of the RFD algorithm specification appeared in 1993, updates and revisions lead to RFC 2439 in 1998 [7].

Mao et al. [8] published in 2002 a paper that studied how the use of RFD, as specified in RFC 2439, can significantly slowdown the convergence times of relatively stable routing entries. This abnormal behavior arises during route withdrawal from the interaction of RFD with "BGP path exploration" (in which in response to path failures or routing policy changes, some BGP routers may try a sequence of transient alternate paths before selecting a new path or declaring the corresponding destination unreachable). Bush et al. [9] summarized the findings of Mao et al. [8] and presented some observational data to illustrate the phenomena. The overall conclusion of this work was to avoid using RFD so that the overall ability of the network to re-converge after an episode of "BGP path exploration" was not needlessly slowed.

More recently, solutions such as the enhanced path vector routing protocol (EPIC) [10] propose to add a forward edge sequence numbers mechanism to annotate the AS paths with additional "path dependency'' information. This information is combined with an enhanced path vector algorithm to limit path exploration and to reduce convergence time in case of failure. EPIC shows significant reduction of convergence time and the number of messages in the fail-down scenario (a part of the network is disconnected from the rest of the network) but only a modest improvement in the fail-over scenario (edges failures without isolation). The main drawback of EPIC is the large amount of extra information stored at the nodes and the increase of the size of messages. Another solution, BGP with Root Cause Notification (RCN) [11] proposes to reduce the BGP convergence delay by announcing the root cause of a link failure location. This solution also offers a significant reduction of the convergence time in the fail-down scenario. However, the convergence time improvement achieved with RCN is modest on the Internet topology compared to legacy BGP (in the fail-over scenario). More advanced techniques such as the recently introduced Path Exploration Damping (PED) [12] augments BGP for selectively damping the propagation of path exploration updates. PED selectively delays and suppresses the propagation of BGP updates that either lengthen an existing AS Path or vary an existing AS-Path without shortening its length.

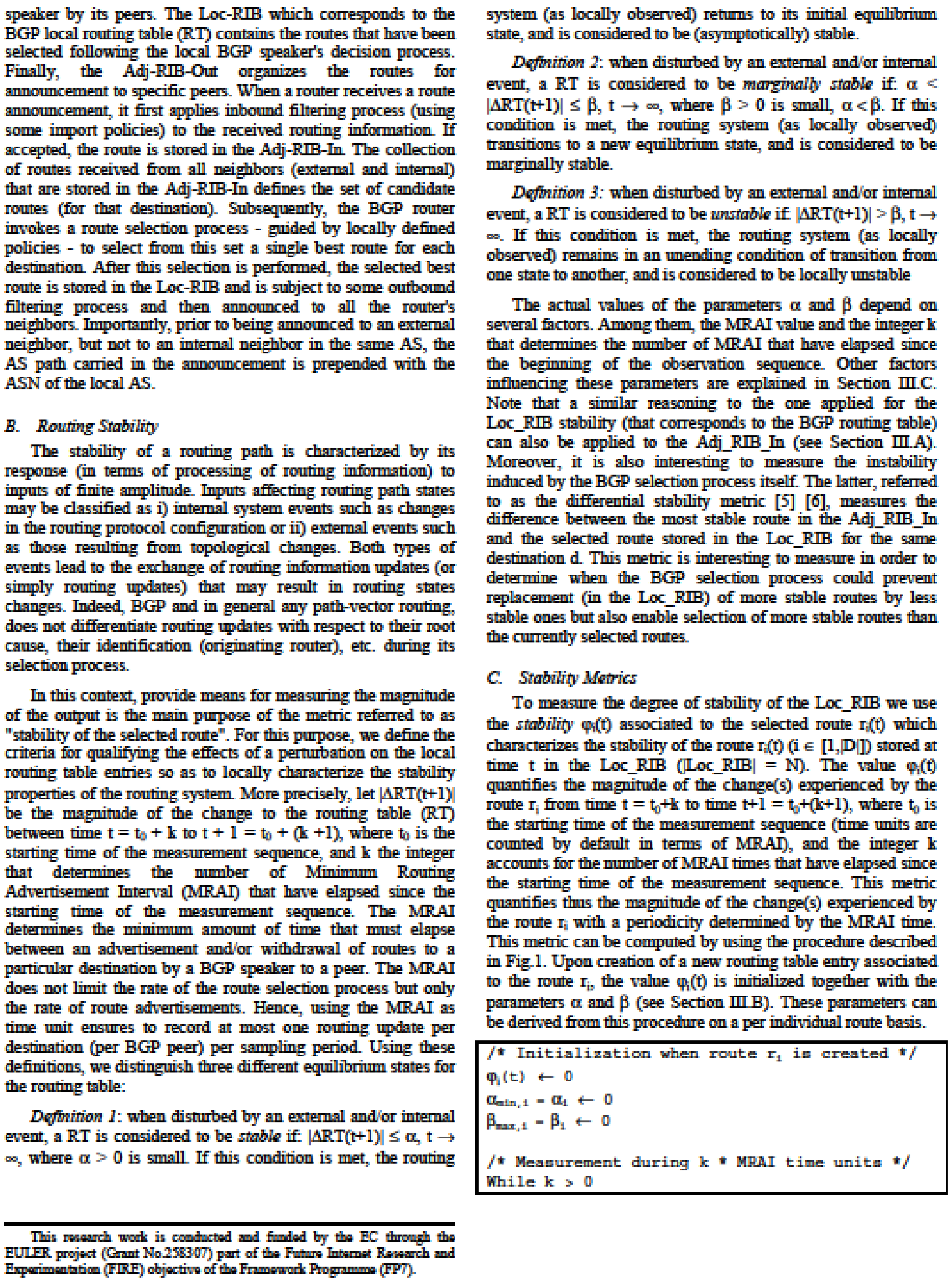
All these approaches try to mitigate instability effects and/or to accelerate convergence after occurrence of a perturbation event, but none of them ask the fundamental question why selecting a route subject to path exploration at first place. The answer is essentially because none of these mechanisms perform rely on the actual quantification of the instability effect and still use network-wide spatial criteria that for AS-Path selection.

III. Routing Stability and Metrics

A. Preliminaries

The autonomous system (AS) topology underlying the routing system is described as a graph G = (V,E), where each vertex (or abstract node) u ∈ V, |V| = n, represents an AS, and each edge e ∈ E, |E| = m, represents a link between an AS pair denoted (u,v), where u, v ∈ V. Each AS comprises a set of physical nodes referred to as routers; the AS representation of the topology combines thus both its partitioning and its abstraction. The subset of physical nodes of interest for this paper comprises the routers running the path-vector algorithm (typically sitting at the periphery of each AS). At each of these routers, a route r per destination d (d ∈ D) is selected and stored as an entry in the local routing table (RT). The total number of entries is denoted by N, i.e., |RT| = N. A route r_i to destination d at time t is defined by $r_i(t) = \{d, (v_k=u, v_{k-1},\ldots,v_0=v), A\}$ with $k > 0 \mid \forall j, k \geq j > 0, \{v_j, v_{j-1}\} \in E$ and $i \in [1,N]$, where $(v_k=u, v_{k-1},\ldots,v_0=v)$ represents the AS-Path, v_{k-1} the next hop of v along the AS-Path from the abstract node u to v, and A its attribute set. Let $P_{(u,v),d}$ denote the set of paths from node u to v towards destination d where each path p(u,v) is of the form $\{(v_k=u, v_{k-1},\ldots,v_0=v), A\}$. A routing information update leads to a change of the AS-Path $(v_k, v_{k-1},\ldots,v_0)$ or an element of its attribute set A. Next, a withdrawal is denoted by an empty AS-Path (ε) and A = ∅: $\{d,\varepsilon,\varnothing\}$. According to the above definition, if there is more than one AS-Path per destination d, they will be considered as multiple distinct routes.

BGP being in the context of this paper the path-vector routing protocol considered; we further detail its storage data structures, referred to as Routing Information Bases (RIBs), used to store its routes $r_i(t)$. At each BGP speaker, the RIB consists of three distinct parts: the Adj-RIB-In, the Loc-RIB, and the Adj-RIB-Out. The Adj-RIB-In contains unprocessed routing information that has been announced to the local BGP



speaker by its peers. The Loc-RIB which corresponds to the BGP local routing table (RT) contains the routes that have been selected following the local BGP speaker's decision process. Finally, the Adj-RIB-Out organizes the routes for announcement to specific peers. When a router receives a route announcement, it first applies inbound filtering process (using some import policies) to the received routing information. If accepted, the route is stored in the Adj-RIB-In. The collection of routes received from all neighbors (external and internal) that are stored in the Adj-RIB-In defines the set of candidate routes (for that destination). Subsequently, the BGP router invokes a route selection process - guided by locally defined policies - to select from this set a single best route for each destination. After this selection is performed, the selected best route is stored in the Loc-RIB and is subject to some outbound filtering process and then announced to all the router's neighbors. Importantly, prior to being announced to an external neighbor, but not to an internal neighbor in the same AS, the AS path carried in the announcement is prepended with the ASN of the local AS.

B. *Routing Stability*

The stability of a routing path is characterized by its response (in terms of processing of routing information) to inputs of finite amplitude. Inputs affecting routing path states may be classified as i) internal system events such as changes in the routing protocol configuration or ii) external events such as those resulting from topological changes. Both types of events lead to the exchange of routing information updates (or simply routing updates) that may result in routing states changes. Indeed, BGP and in general any path-vector routing, does not differentiate routing updates with respect to their root cause, their identification (originating router), etc. during its selection process.

In this context, provide means for measuring the magnitude of the output is the main purpose of the metric referred to as "stability of the selected route". For this purpose, we define the criteria for qualifying the effects of a perturbation on the local routing table entries so as to locally characterize the stability properties of the routing system. More precisely, let $|\Delta RT(t+1)|$ be the magnitude of the change to the routing table (RT) between time $t = t_0 + k$ to $t + 1 = t_0 + (k +1)$, where t_0 is the starting time of the measurement sequence, and k the integer that determines the number of Minimum Routing Advertisement Interval (MRAI) that have elapsed since the starting time of the measurement sequence. The MRAI determines the minimum amount of time that must elapse between an advertisement and/or withdrawal of routes to a particular destination by a BGP speaker to a peer. The MRAI does not limit the rate of the route selection process but only the rate of route advertisements. Hence, using the MRAI as time unit ensures to record at most one routing update per destination (per BGP peer) per sampling period. Using these definitions, we distinguish three different equilibrium states for the routing table:

Definition 1: when disturbed by an external and/or internal event, a RT is considered to be *stable* if: $|\Delta RT(t+1)| \le \alpha$, $t \to \infty$, where $\alpha > 0$ is small. If this condition is met, the routing system (as locally observed) returns to its initial equilibrium state, and is considered to be (asymptotically) stable.

Definition 2: when disturbed by an external and/or internal event, a RT is considered to be *marginally stable* if: $\alpha < |\Delta RT(t+1)| \le \beta$, $t \to \infty$, where $\beta > 0$ is small, $\alpha < \beta$. If this condition is met, the routing system (as locally observed) transitions to a new equilibrium state, and is considered to be marginally stable.

Definition 3: when disturbed by an external and/or internal event, a RT is considered to be *unstable* if: $|\Delta RT(t+1)| > \beta$, $t \to \infty$. If this condition is met, the routing system (as locally observed) remains in an unending condition of transition from one state to another, and is considered to be locally unstable

The actual values of the parameters α and β depend on several factors. Among them, the MRAI value and the integer k that determines the number of MRAI that have elapsed since the beginning of the observation sequence. Other factors influencing these parameters are explained in Section III.C. Note that a similar reasoning to the one applied for the Loc_RIB stability (that corresponds to the BGP routing table) can also be applied to the Adj_RIB_In (see Section III.A). Moreover, it is also interesting to measure the instability induced by the BGP selection process itself. The latter, referred to as the differential stability metric [5] [6], measures the difference between the most stable route in the Adj_RIB_In and the selected route stored in the Loc_RIB for the same destination d. This metric is interesting to measure in order to determine when the BGP selection process could prevent replacement (in the Loc_RIB) of more stable routes by less stable ones but also enable selection of more stable routes than the currently selected routes.

C. *Stability Metrics*

To measure the degree of stability of the Loc_RIB we use the *stability* $\varphi_i(t)$ associated to the selected route $r_i(t)$ which characterizes the stability of the route $r_i(t)$ ($i \in [1,|D|]$) stored at time t in the Loc_RIB (|Loc_RIB| = N). The value $\varphi_i(t)$ quantifies the magnitude of the change(s) experienced by the route r_i from time $t = t_0+k$ to time $t+1 = t_0+(k+1)$, where t_0 is the starting time of the measurement sequence (time units are counted by default in terms of MRAI), and the integer k accounts for the number of MRAI times that have elapsed since the starting time of the measurement sequence. This metric quantifies thus the magnitude of the change(s) experienced by the route r_i with a periodicity determined by the MRAI time. This metric can be computed by using the procedure described in Fig.1. Upon creation of a new routing table entry associated to the route r_i, the value $\varphi_i(t)$ is initialized together with the parameters α and β (see Section III.B). These parameters can be derived from this procedure on a per individual route basis.

```
/* Initialization when route $r_i$ is created */
$\varphi_i(t) \leftarrow 0$
$\alpha_{min,i} = \alpha_i \leftarrow 0$
$\beta_{max,i} = \beta_i \leftarrow 0$

/* Measurement during k * MRAI time units */
While k > 0
```

This research work is conducted and funded by the EC through the EULER project (Grant No.258307) part of the Future Internet Research and Experimentation (FIRE) objective of the Framework Programme (FP7).

```
if $\Delta r_i(t+1) \neq 0$
   /* $r_i$ experiences an AS-Path change
      or $r_i$ experiences an attribute change */
then $\varphi_i(t+1) \leftarrow \varphi_i(t) + 1$
     $\beta_i \leftarrow \varphi_i(t+1)$
     if $\beta_{max,i} < \beta_i$ then $\beta_{max,i} \leftarrow \beta_i$
     end if
else /* $r_i$ experiences no change: $\Delta r_i(t+1)=0$ */
     if $\varphi_i(t) > 0$
     then $\varphi_i(t+1) \leftarrow \varphi_i(t) - 1$
          $\alpha_i \leftarrow \varphi_i(t+1)$
          if $\alpha_i > \alpha_{min,i}$ then $\alpha_{min,i} \leftarrow \alpha_i$
          end if
     else $\varphi_i(t+1) \leftarrow 0$
     end if
end if
$k \leftarrow k - 1$
end k loop
```

Figure 1. Stability of individual routes

D. *Forwarding Stability*

The RADAR tool records sequences of IP addresses corresponding to the routers traversed by the forwarding path and not the AS-path (as provided by the RouteView tool). Thus, before computing its stability metrics, each forwarding path needs to be associated to the corresponding AS number sequence (corresponding to the routing path) following the procedure documented in Section IV.B. Then the stability of each forwarding path can be computed following the algorithm described in Fig.1. As the computation algorithm is applied to the AS sequence instead of the IP address sequence of each forwarding path, the stability metric identifies changes "inter-AS" changes in the forwarding path (not "intra-AS" changes). Identification of additional "intra-AS" changes is left for future study.

IV. Datasets and Processing Method

In this section we describe the processing method applied to the results of the stability metric computation as obtained from the application of the algorithm described in Section III.

A. *Datasets*

The computation of the stability of the routing paths relies on the processing BGP update messages as collected by the RouteViews project (www.routeviews.org). The computation of the stability of the forwarding paths makes use of the forwarding paths as recorded by the RADAR tool [13].

1) *Routing Path Dataset*

RouteViews [14] is a project founded and sponsored by the University of Oregon which consists in a set of BGP routers distributed worldwide. The BGP routing information collected by these routers is stored in BGP feeds. The BGP datasets obtained from these routers can be openly accessed by anyone, interested or involved in the field of Internet (routing) research. This information has led to various noticeable studies including those conducted in [15] and [16]. More precisely, the BGP datasets obtained from RouteViews contain the BGP routing information a monitored router receives from its neighboring BGP speakers. That is basically each neighbor route (with its route attributes) to each address prefix the neighbor has knowledge about. The data obtained from these monitored routers comprise i) the complete Routing Information Base (RIB) entries updated every two hours and ii) the received updates from the peer ASs separated in files every 15 minutes. The format used to encode the records in these files is MRT [17].

The monitored router is route-views.wide.routeviews.org in order to facilitate the association with the forwarding path dataset described in Section IV.B. We use the tool developed in [5] [6] to process all the BGP data collected from this RouteViews router.

2) *Forwarding Path Dataset*

The measurements carried out by RADAR are traceroute-like probes initiated from a set of monitoring nodes. Such probes target a large set of IP address prefixes and end-hosts distributed across the Internet. Based on these measures, the RADAR tool builds ego-centered views of the forwarding topology (in other terms, the initiating router collects traces along the forwarding paths that it probes). A subset of the forwarding paths traced by the RADAR probes corresponds expectedly to the routers monitored by RouteViews; consequently, a subset of the monitored AS-Paths is also monitored by RADAR.

B. *Datasets Pre-Processing*

The first step to relate the stability of a given forwarding path to the corresponding routing path is to find the possible association between the dataset records provided by the RADAR tool (forwarding paths) and RouteViews (routing paths). Indeed, the data provided by RADAR are sequences of IP addresses while the BGP routes as provided by the RouteViews datasets are sequences of AS numbers (AS-Path). Finding association (or matching) between IP forwarding paths to AS routing paths is thus required. Performing this operation can be obtained by executing the Whois protocol [18]. Whois is a TCP-based transaction-oriented query/response protocol that is widely used for querying databases that store the registered users or assignees of an Internet resource, such as a domain name, an IP address prefix, or an AS. We have used the Whois-based web tool provided by the Team Cymru (http://www.team-cymru.org/). This specific tool takes as input a file containing IP addresses and translates them into AS numbers as output of the tool developed in C++ programming language. In total, each sample includes a bit less than 1000 forwarding path - routing path sequences. It is important to mention that performing association between pairs of forwarding and routing paths (per destination) does not require the full identification of data sequences before association but only that a given forwarding path (IP address sequence) can be associated unambiguously to a given routing path (AS number sequence). Hence, the identification problem could be limited to a specific subset of the total number of pairs (those experiencing instability). Moreover, this method working by association is much simpler compared to the one that would require full mapping of IP addresses (forwarding paths) and AS (routing paths) before computation and analysis.

On the other hand, the interval of the measurement applied by the RADAR tool and the MRAI time interval used to process the BGP routes (obtained from RouteViews) are basically different. In RADAR, each measurement round takes approximately 4 minutes and 10 minutes elapses between the end of a given round and the beginning of the next one. We thus run two different sets of execution with the routes obtained from the RouteViews data.

- MRAI time interval: The first set of executions uses the actual BGP UPDATE message time interval as determined by the MRAI time; the stability of the routing path is therefore computed (per destination) according to the real BGP UPDATE message period as regulated by the MRAI. Moreover, the value of the forwarding path stability is not computed at this granularity but assumed to remain constant at the value computed at the beginning of each 4 minute interval.
- RADAR time interval: the second set of executions relies on the RADAR iteration (around each 10 minutes). In this case, the routing path stability variation(s) are accounted following the RADAR time interval; thus, in this case the timing at which the stability metric computation is performed is driven by the RADAR tool.

After having associated the forwarding paths to the routing paths (i.e., produce pairs of rouand scaled their measurement intervals, one can then i) compute the stability metric (as defined in Section III) for each routing path using the collected BGP datasets, ii) compute the stability metric (as defined in Section III) for the corresponding forwarding path using the collected RADAR datasets, iii) derive the associated statistics and evolution over time.

C. Processing Method

Based on this set of pairs of sequences, a first characterization of the observed instability would consist in determining whether instability events can be detected or not at the forwarding and/or the routing path (FP and RP, respectively) level. For this purpose, these pairs can be then grouped (or classified) into 4 subsets (or classes) following the type of experienced event:

1. `(FP_Stable,RP_Stable)`: obviously such pair does not require any further processing or analysis.
2. `(FP_Stable,RP_Unstable)`: each pair comprised as part of this class translates routing path instability with forwarding path stability.
3. `(FP_Unstable,RP_Stable)`: each pair comprised as part of this class translates routing path stability without forwarding path stability.
4. `(FP_Unstable,RP_Unstable)`: each pair comprised as part of this class requires identification if a common segment is at the origin of the instability (thus the AS Path -IP address mapping is required to determine whether there is a common origin to the observed instability).

Over all pairs belonging to each class, we record the minimum and the maximum value of the stability metric in addition to the computation of the average and variance of the stability metric. Pairs part of classes (Class_2) and (Class_3) are also interesting to analyze because they translate routing path instability without forwarding path instability and vice-versa; identifying the origin of the instability for the pairs belonging to corresponding classes can be performed in a second phase of analysis (as more costly).

After a couple of initial executions performed over the whole duration of the measurement period, it became clear that it was simply not possible to classify all routing path - forwarding path pairs by means of a single discriminant. Indeed, some of these pairs can exhibit multiple patterns during the measurement period and capturing this behavior under a single value was not reproducing sufficient information. Instead, we adapted the procedure and count the number of events labeled as `(FP_Stable,RP_Stable)`, `(FP_Stable,RP_Unstable)`, `(FP_Unstable, RP_Stable)`, and `(FP_Unstable, RP_Unstable)` observed for each forwarding-routing path pair. We then derive a dominant/main trend corresponding to the label with the maximum number of counts and a sub-trend. We also count for each pair the duration (in MRAI time units) associated to the occurrence of the events that are determined by the above-mentioned classes. This additional processing enables to derive for each pair a dominant and a sub-trend with respect to their duration. Indeed, certain pair may have a very few number of counts but certain of them may be very long.

V. Results and Analysis

Table I summarizes the classification of the routing path (RP) - forwarding path (FP) pairs using the following class labels `(FP_unstable, RP_unstable)`, `(FP_unstable, RP_stable)`, and `(FP_stable, RP_unstable)`. The second column indicates the absolute number and the percentage of paths per class for which at least one instability event has been observed. The third column provides the maximum number of measurement intervals over which the corresponding behavior has been observed together with the median value.

Table I: Classification of FP-RP pairs

Label	Number and Percent of Pairs	Max Count - Median
FP unstable - RP unstable	517 - 54%	40 - 2
FP unstable - RP stable	915 - 96%	223 - 47
FP stable - RP unstable	182 - 19%	117 - 2

Table II details the observed dominant behavior/trend and the sub-trend; each computed as follows: a score of 1 is assigned to the dominant behavior and 0 to the others. In case a balance is observed between two (three) classes, a score of 0.5 (0.33) is assigned to each of them. As it can be observed from this table, the majority of the routing path (RP) - forwarding path (FP) pairs falls in the (FP_unstable,RP_stable) class. The latter determines the dominant behavior, i.e., the most representative behavior. The total number of instability events observed for about 95% of the pairs results from forwarding path instability. Few pairs (less than 4%) are labeled as (FP_unstable, RP_unstable), meaning that only a small fraction of the routing paths experiencing instability events corresponds to forwarding path instability. The second trend indicates that for about 50% of the pairs the observed instability result from both forwarding and routing path instability.

TABLE II: TREND ANALYSIS

Main trend	Number of Pairs	Score
FP unstable - RP unstable	36	32
FP unstable - RP stable	912	906
FP stable - RP unstable	15	12
Second trend	**Number of Pairs**	**Score**
FP unstable - RP unstable	474	444
FP unstable - RP stable	3	3
FP stable - RP unstable	58	12

Figure 2 plots the percentage of the observed instability events (i.e., either the forwarding or the routing path would be unstable) over the entire measurement period in the form of a cumulative distribution function (CDF). From this figure, the following observations can be drawn. The majority of the pairs (around 60%) are labeled as (FP_stable, RP_stable), i.e., the instability events observed for each of these pairs account for less than 10% of the observed events; moreover, for around 75% of the pairs, the percentage of observed instability events is less or equal to 20% of the total number of events. The latter percentage increases to 50% when reaching about 87% of the pairs (i.e., for only 13% of the pairs, the instability events experienced is higher or equal to 50% of total number of events).

We also computed the total duration (counted in MRAI time units) of the instability events observed for each pair in order to distinguish the number of transitions to instability events from their actual duration. From this computation, we determine that the instability events observed for about 99% of the pairs results in majority from forwarding path instability. More precisely, the dominant instability behavior over time is characterized by a majority of pairs belonging to the (FP_unstable, RP_stable) class. More generally, the results obtained in terms of duration measurement tend to enforce the main trend observed from Table II. However, they do not confirm those observed for the second trend: number of (FP_unstable,RP_unstable) pairs 145 vs 474 and number of (FP_stable,RP_ unstable) pairs 793 vs 58. This seems to imply that for a majority (about 85%) of the pairs the second order temporal effect is dominated by routing path instability (without forwarding path instability).

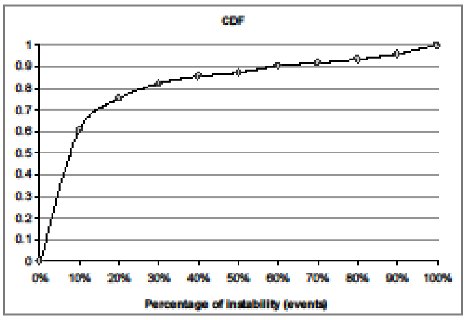


Figure 2. Cumulative Distribution Function vs Perc. of instability (events)

Figure 3 plots the percentage of time during which the instability events have been observed (i.e., either the forwarding or the routing path would be unstable) over the entire measurement period in the form of a cumulative distribution function (CDF). The following observations can be drawn from this figure. The majority of the pairs (about 60%) can be labeled as (FP_stable, RP_stable), i.e., the cumulated time of instability accounts for less than 10% of the total duration (i.e., during the remaining 90% of the time the observed events are labeled as (FP_stable, RP_stable)). Moreover, for about 75% of the pairs, the observed instability events account for up to 20% of the total duration (i.e., during 80% of the time the observed events are labeled as (FP_stable, RP_stable)). The latter percentage increases to 50% when reaching about 87% of the pairs.

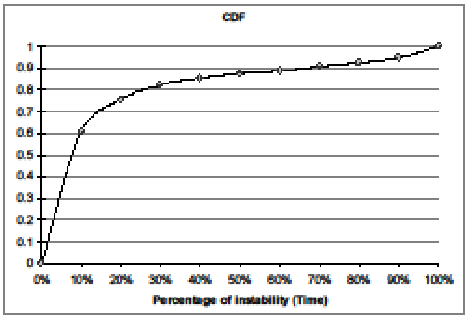
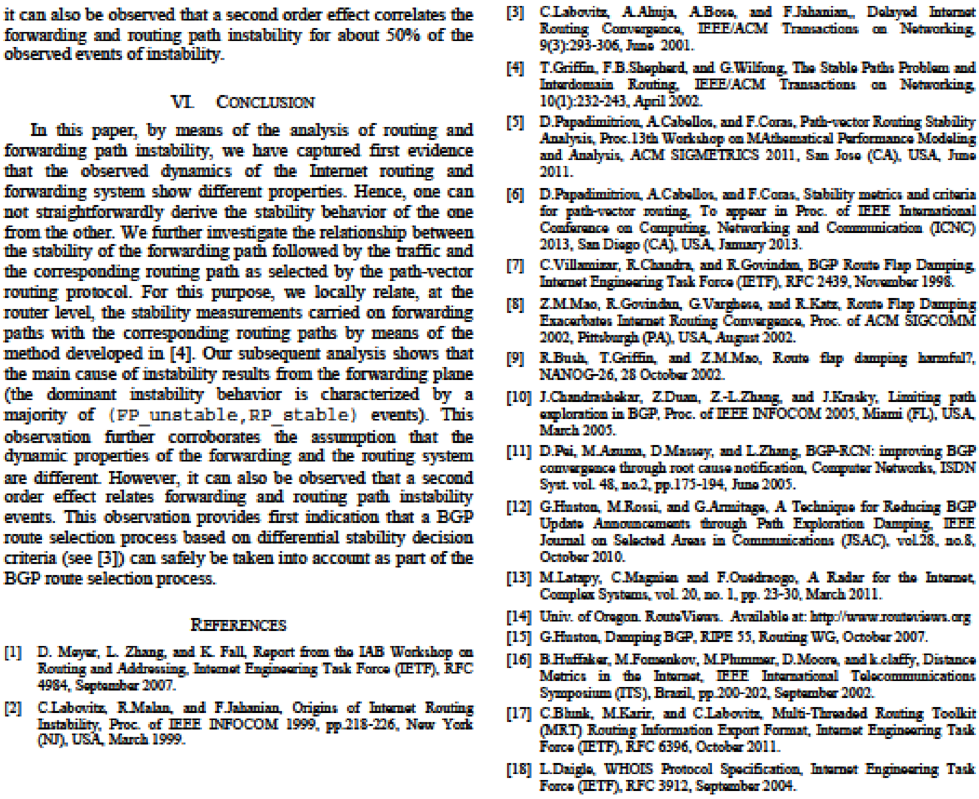


Figure 3. Cumulative Distribution Function vs Perc. of instability (time)

These observations combined with the fact that the main cause of instability results from the forwarding plane corroborates the assumption that the dynamic properties of the forwarding and the routing system are different. Henceforth, it is impossible to derive one behavior from the other. Moreover,



it can also be observed that a second order effect correlates the forwarding and routing path instability for about 50% of the observed events of instability.

VI. Conclusion

In this paper, by means of the analysis of routing and forwarding path instability, we have captured first evidence that the observed dynamics of the Internet routing and forwarding system show different properties. Hence, one can not straightforwardly derive the stability behavior of the one from the other. We further investigate the relationship between the stability of the forwarding path followed by the traffic and the corresponding routing path as selected by the path-vector routing protocol. For this purpose, we locally relate, at the router level, the stability measurements carried on forwarding paths with the corresponding routing paths by means of the method developed in [4]. Our subsequent analysis shows that the main cause of instability results from the forwarding plane (the dominant instability behavior is characterized by a majority of `(FP_unstable,RP_stable)` events). This observation further corroborates the assumption that the dynamic properties of the forwarding and the routing system are different. However, it can also be observed that a second order effect relates forwarding and routing path instability events. This observation provides first indication that a BGP route selection process based on differential stability decision criteria (see [3]) can safely be taken into account as part of the BGP route selection process.

References

[1] D. Meyer, L. Zhang, and K. Fall, Report from the IAB Workshop on Routing and Addressing, Internet Engineering Task Force (IETF), RFC 4984, September 2007.

[2] C.Labovitz, R.Malan, and F.Jahanian, Origins of Internet Routing Instability, Proc. of IEEE INFOCOM 1999, pp.218-226, New York (NJ), USA, March 1999.

[3] C.Labovitz, A.Ahuja, A.Bose, and F.Jahanian,, Delayed Internet Routing Convergence, IEEE/ACM Transactions on Networking, 9(3):293-306, June 2001.

[4] T.Griffin, F.B.Shepherd, and G.Wilfong, The Stable Paths Problem and Interdomain Routing, IEEE/ACM Transactions on Networking, 10(1):232-243, April 2002.

[5] D.Papadimitriou, A.Cabellos, and F.Coras, Path-vector Routing Stability Analysis, Proc.13th Workshop on MAthematical Performance Modeling and Analysis, ACM SIGMETRICS 2011, San Jose (CA), USA, June 2011.

[6] D.Papadimitriou, A.Cabellos, and F.Coras, Stability metrics and criteria for path-vector routing, To appear in Proc. of IEEE International Conference on Computing, Networking and Communication (ICNC) 2013, San Diego (CA), USA, January 2013.

[7] C.Villamizar, R.Chandra, and R.Govindan, BGP Route Flap Damping, Internet Engineering Task Force (IETF), RFC 2439, November 1998.

[8] Z.M.Mao, R.Govindan, G.Varghese, and R.Katz, Route Flap Damping Exacerbates Internet Routing Convergence, Proc. of ACM SIGCOMM 2002, Pittsburgh (PA), USA, August 2002.

[9] R.Bush, T.Griffin, and Z.M.Mao, Route flap damping harmful?, NANOG-26, 28 October 2002.

[10] J.Chandrashekar, Z.Duan, Z.-L.Zhang, and J.Krasky, Limiting path exploration in BGP, Proc. of IEEE INFOCOM 2005, Miami (FL), USA, March 2005.

[11] D.Pei, M.Azuma, D.Massey, and L.Zhang, BGP-RCN: improving BGP convergence through root cause notification, Computer Networks, ISDN Syst. vol. 48, no.2, pp.175-194, June 2005.

[12] G.Huston, M.Rossi, and G.Armitage, A Technique for Reducing BGP Update Announcements through Path Exploration Damping, IEEE Journal on Selected Areas in Communications (JSAC), vol.28, no.8, October 2010.

[13] M.Latapy, C.Magnien and F.Ouedraogo, A Radar for the Internet, Complex Systems, vol. 20, no. 1, pp. 23-30, March 2011.

[14] Univ. of Oregon. RouteViews. Available at: http://www.routeviews.org

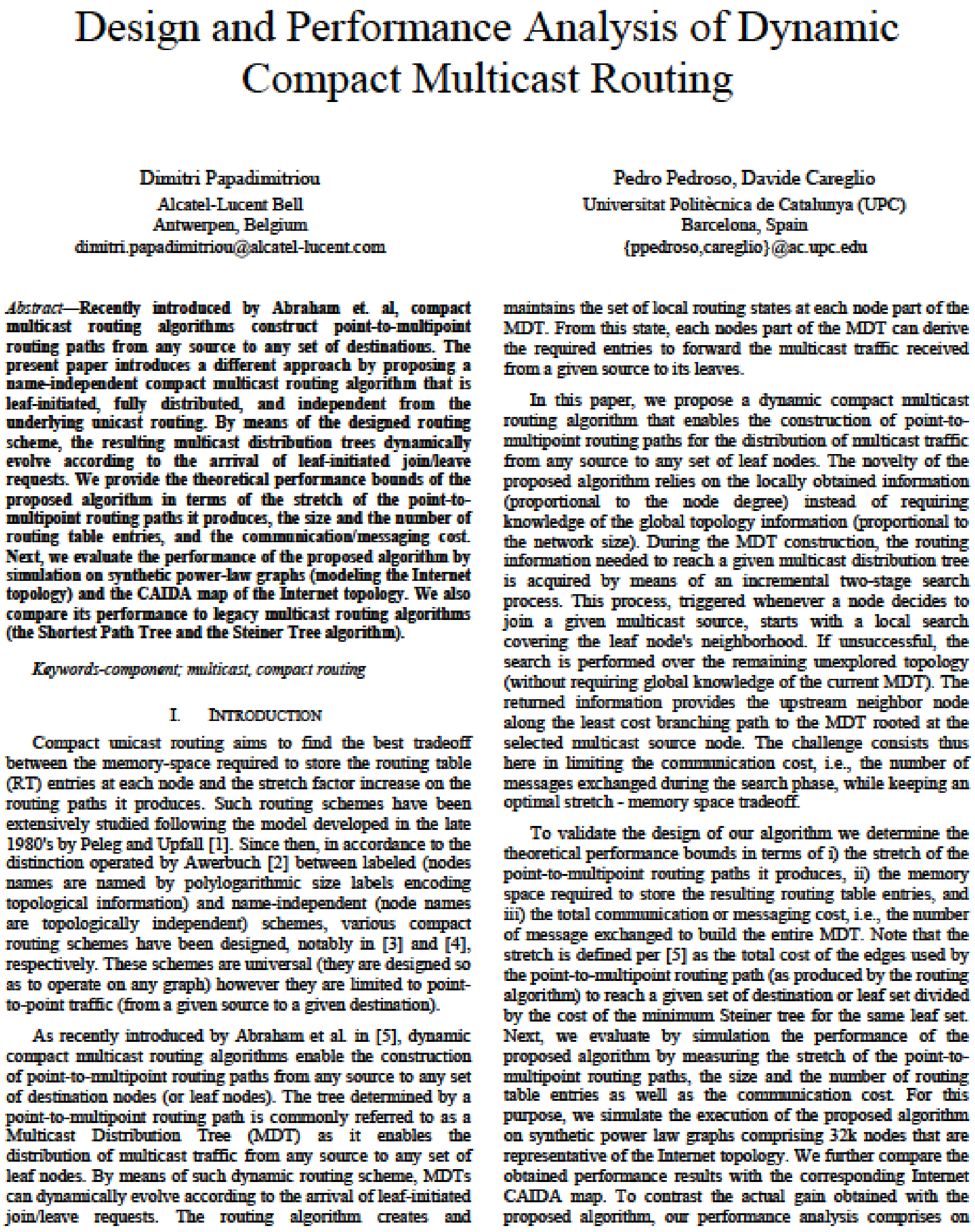
[15] G.Huston, Damping BGP, RIPE 55, Routing WG, October 2007.

[16] B.Huffaker, M.Fomenkov, M.Plummer, D.Moore, and k.claffy, Distance Metrics in the Internet, IEEE International Telecommunications Symposium (ITS), Brazil, pp.200-202, September 2002.

[17] C.Blunk, M.Karir, and C.Labovitz, Multi-Threaded Routing Toolkit (MRT) Routing Information Export Format, Internet Engineering Task Force (IETF), RFC 6396, October 2011.

[18] L.Daigle, WHOIS Protocol Specification, Internet Engineering Task Force (IETF), RFC 3912, September 2004.

Annex 4: Paper "Design and Performance Analysis of Dynamic Compact Multicast Routing"



Design and Performance Analysis of Dynamic Compact Multicast Routing

Dimitri Papadimitriou
Alcatel-Lucent Bell
Antwerpen, Belgium
dimitri.papadimitriou@alcatel-lucent.com

Pedro Pedroso, Davide Careglio
Universitat Politècnica de Catalunya (UPC)
Barcelona, Spain
{ppedroso,careglio}@ac.upc.edu

Abstract—**Recently introduced by Abraham et. al, compact multicast routing algorithms construct point-to-multipoint routing paths from any source to any set of destinations. The present paper introduces a different approach by proposing a name-independent compact multicast routing algorithm that is leaf-initiated, fully distributed, and independent from the underlying unicast routing. By means of the designed routing scheme, the resulting multicast distribution trees dynamically evolve according to the arrival of leaf-initiated join/leave requests. We provide the theoretical performance bounds of the proposed algorithm in terms of the stretch of the point-to-multipoint routing paths it produces, the size and the number of routing table entries, and the communication/messaging cost. Next, we evaluate the performance of the proposed algorithm by simulation on synthetic power-law graphs (modeling the Internet topology) and the CAIDA map of the Internet topology. We also compare its performance to legacy multicast routing algorithms (the Shortest Path Tree and the Steiner Tree algorithm).**

Keywords-component; multicast, compact routing

I. Introduction

Compact unicast routing aims to find the best tradeoff between the memory-space required to store the routing table (RT) entries at each node and the stretch factor increase on the routing paths it produces. Such routing schemes have been extensively studied following the model developed in the late 1980's by Peleg and Upfall [1]. Since then, in accordance to the distinction operated by Awerbuch [2] between labeled (nodes names are named by polylogarithmic size labels encoding topological information) and name-independent (node names are topologically independent) schemes, various compact routing schemes have been designed, notably in [3] and [4], respectively. These schemes are universal (they are designed so as to operate on any graph) however they are limited to point-to-point traffic (from a given source to a given destination).

As recently introduced by Abraham et al. in [5], dynamic compact multicast routing algorithms enable the construction of point-to-multipoint routing paths from any source to any set of destination nodes (or leaf nodes). The tree determined by a point-to-multipoint routing path is commonly referred to as a Multicast Distribution Tree (MDT) as it enables the distribution of multicast traffic from any source to any set of leaf nodes. By means of such dynamic routing scheme, MDTs can dynamically evolve according to the arrival of leaf-initiated join/leave requests. The routing algorithm creates and maintains the set of local routing states at each node part of the MDT. From this state, each nodes part of the MDT can derive the required entries to forward the multicast traffic received from a given source to its leaves.

In this paper, we propose a dynamic compact multicast routing algorithm that enables the construction of point-to-multipoint routing paths for the distribution of multicast traffic from any source to any set of leaf nodes. The novelty of the proposed algorithm relies on the locally obtained information (proportional to the node degree) instead of requiring knowledge of the global topology information (proportional to the network size). During the MDT construction, the routing information needed to reach a given multicast distribution tree is acquired by means of an incremental two-stage search process. This process, triggered whenever a node decides to join a given multicast source, starts with a local search covering the leaf node's neighborhood. If unsuccessful, the search is performed over the remaining unexplored topology (without requiring global knowledge of the current MDT). The returned information provides the upstream neighbor node along the least cost branching path to the MDT rooted at the selected multicast source node. The challenge consists thus here in limiting the communication cost, i.e., the number of messages exchanged during the search phase, while keeping an optimal stretch - memory space tradeoff.

To validate the design of our algorithm we determine the theoretical performance bounds in terms of i) the stretch of the point-to-multipoint routing paths it produces, ii) the memory space required to store the resulting routing table entries, and iii) the total communication or messaging cost, i.e., the number of message exchanged to build the entire MDT. Note that the stretch is defined per [5] as the total cost of the edges used by the point-to-multipoint routing path (as produced by the routing algorithm) to reach a given set of destination or leaf set divided by the cost of the minimum Steiner tree for the same leaf set. Next, we evaluate by simulation the performance of the proposed algorithm by measuring the stretch of the point-to-multipoint routing paths, the size and the number of routing table entries as well as the communication cost. For this purpose, we simulate the execution of the proposed algorithm on synthetic power law graphs comprising 32k nodes that are representative of the Internet topology. We further compare the obtained performance results with the corresponding Internet CAIDA map. To contrast the actual gain obtained with the proposed algorithm, our performance analysis comprises on one hand, the comparison between the results obtained for the proposed algorithm and those produced by the Abraham compact multicast routing scheme [5]. On the other hand, two reference schemes, the Shortest Path Tree (SPT) and the Steiner Tree (ST) algorithm are used to compare the performance of the proposed algorithm obtained when running over the same topologies.

This paper is organized as follows. Section II confronts our contribution to prior work on compact multicast routing while motivating the proposed approach compared to legacy multicast routing schemes. We detail the design of the proposed compact multicast routing algorithm in Section III and in Section IV, we provide the theoretical performance bounds in terms of the stretch of point-to-multipoint routing paths it produces, the memory space required for storing the routing table entries, and the communication cost. Section V analyses the performance results obtained by simulation over synthetic network topologies and CAIDA maps of 32k nodes. This section also compares the performance of the proposed algorithm with those realizable with the Abraham routing scheme. Finally, Section VI concludes this paper.

II. Prior Work and Our Contribution

Prior work on compact multicast routing is as far as our knowledge goes mainly concentrated around the schemes developed in the seminal paper authored by Abraham in 2009 [5]. One of the reasons we can advocate is that despite the amount of research work dedicated to compact unicast routing, current schemes are not yet able to efficiently cope with the dynamics of large scale networks. Therefore, running compact multicast routing independently of the underlying unicast routing system would be beneficial. This independence is even the fundamental concept underlying multicast routing schemes such as Protocol Independent Multicast [6]. Nevertheless, we also observe that the scaling problems already faced when multicast routing received main attention from the research community, remain largely unaddressed since so far. Indeed, multicast currently operates as an addressable IP overlay (Class D group addresses) on top of unicast routing topology, leaving up to an order of 100millions of multicast routing table entries. Hence, the need to enable point-to-multipoint routing paths (for bandwidth saving purposes) while keeping multicast addressing at the edges of the network and build shared but selective trees inside the network. Indeed, in our approach, multicast forwarding relies on local port information only. Thus memory capacity savings comes from i) keeping 1:N relationship between network edge node and the number of multicast groups and ii) local port-based addressing for the local processing of multicast traffic. Further, we argue that compact multicast routing, by providing the best memory-space vs stretch tradeoff, can possibly address these scaling challenges without requiring deployment of a compact unicast routing scheme.

A. *Preliminaries*

Consider a network topology modeled by an undirected graph $G = (V,E,c)$ where the set V, $|V| = n$, represents the finite set of nodes or vertices (all being multicast capable), the set E, $|E| = m$, represents the finite set of links or edges, and c a non-negative cost function $c: E \rightarrow Z+$ that associates a non-negative cost $c(u,v)$ to each link $(u,v) \in E$. For $u, v \in V$, let $c(u,v)$ denote the cost of the path $p(u,v)$ from u to v in G, where the cost of a path is defined as the sum of the costs along its edges. Let S, $S \subset V$, be the finite set of source nodes, and $s \in S$ denote a source node. Let D, $D \subseteq V \backslash \{S\}$, be the finite set of all possible destination nodes that can join a multicast source s, and $d \in D$ denote a destination (or leaf) node. A *multicast distribution tree* $T_{s,M}$ is defined as an acyclic connected sub-graph of G, i.e., a tree rooted at source $s \in S$ with leaf node set M, $M \subseteq D$.

B. *Prior Work*

We outline the dynamic compact multicast routing scheme for join-only events scheme as designed by Abraham et al. (part of their seminal work [5]). In Section V, we provide a detailed comparison between the performance results obtained with our routing scheme and their approach. The Abraham scheme relies on the off-line construction of a bundle $\mathcal{B}_k$ of sparse covers $TC_{k,2i}$, defined as $\mathcal{B}_k = \{TC_{k,2i}(G) \mid i \in I\}$ with $k = \log(n)$. Sparse covers are grown from a set of center nodes $c(T_i(v))$ located at distance at most $k2^i$ from node v, where $T_i(v)$ denotes the tree in the collection of rooted trees $TC_{k,2i}(G)$ that contains the ball $B(v,2^i)$. For each $i \in I$ and $T \in TC_{k,2i}(G)$, the center node $c(T(v))$ of each node $v \in T$ stores the labels of all nodes[1] contained in the ball $B(v,2^i)$, the ball centered on node v of radius 2^i. Further, the SPlabel(v) stores the label $\lambda(T,c(T))$ for each $T \in \mathcal{B}(v)$, defined as set of all covers T in the bundle $\mathcal{B}_k$ such that $v \in T$. In addition, each node $v \in V$ stores the tree routing information $\mu(T,v)$ for all the trees in its own label SPlabel(v). When a leaf node u desires to join an MDT, it first determines whether or not one of the MDT nodes is already included in its local tree routing information table. If this is the case, it sends the join request to the center node $c(T_i(v))$ with minimum degree cover $i \in I$ that is associated to that MDT node v. The center node $c(T_i(v))$ then passes the label $\mu(T_i(v),u)$ so that the selected MDT node v can forward the multicast traffic to the newly joining leaf node u (without further propagating this label to the source node s). Otherwise (some the leaf node covers define an empty intersection with the MDT), the leaf node u with SPLabel(u) queries the source node s to obtain the set of MDT nodes it currently includes. Among all index $i \in I$, it then selects the tree $T_{i*}(v)$ whose intersection with its bundle $\mathcal{B}(u)$ is minimum. Once the node, say v, part of this intersection is selected $(T_{i*}(v) \in \mathcal{B}(u))$, leaf node u directs the join request to the associated center node $c(T_{i*}(v))$. The latter passes a label $\mu(T_{i*}(v),u)$ so that the selected MDT node v can forward the incoming multicast traffic to the newly joining leaf node u. In order for the source node s to reach node u, node v has to propagate the tuple $[v,c(T_{i*}(v)),\mu(T_{i*}(v),u)]$ to source s. The leaf node u updates all nodes covered by its balls $B(u,2^i)$ to allow them joining the MDT at node u.

[1] For simplicity, we present here the label-dependent variant of the scheme. In the name-independence version, center nodes store label mappings from names to nodes.

Compared to the Abraham scheme [5], our name-independent compact multicast routing algorithm is also i) *leaf-initiated* since join requests are initiated by the leaf nodes; however, contrary to the Abraham scheme it operates without requiring prior local dissemination of the node set already part of the MDT or keeping specialized nodes informed about nodes that have joined the MDT, and ii) *dynamic* since requests are processed on-line as they arrive without re-computing and/or re-building the MDT. Moreover, our proposed algorithm is iii) *distributed* since transit nodes process homogeneously the incoming requests to derive the least cost branching path to the MDT without requiring any centralized processing by the root of the MDT or any specialized processing by means of pre-determined center nodes, and iv) *independent* of any underlying sparse cover construction grown from a set of center nodes (which induce node specialization driving the routing functionality): the local knowledge of the cost to direct neighbor nodes is sufficient for the proposed algorithm to properly operate. It is important to emphasize that the sparse cover underlying the Abraham scheme is constructed off-line and requires global knowledge of the network topology to properly operate.

C. *Our Contribution*

The objective of the proposed algorithm is to minimize the routing table sizes of each node part of the MDT at the expense of i) routing the packets on point-to-multipoint paths with relative small deviation compared to the optimal stretch obtained by the Steiner Tree (ST) algorithm, and ii) higher communication cost compared to the Shortest Path Tree (SPT) algorithm. For this purpose, the proposed algorithm reduces the local storage of routing information by keeping only direct neighbor-related entries rather than tree structures (as in ST) or network graph entries (as in both SPT and ST). In other terms, the novelty of the proposed algorithm is on requiring maintenance of only local topology information while providing the least cost next hop during the MDT construction. That is, our algorithm does not rely on the knowledge of the global topology information or involve the construction of global network structures such as sparse covers. The information needed to reach a given multicast source is acquired by means of a two-stage search process that returns the upstream node along the least cost branching path to the MDT sourced at s. This process is triggered whenever a node decides to join a given multicast source s, root of the MDT. After a node becomes member of a MDT, a multicast routing entry is dynamically created and stored in the local tree information base (TIB). From these routing table entries, multicast forwarding entries are locally instantiated.

As stated before, the reduction in memory space consumed by the routing table entries results however in higher communication cost compared to the reference algorithms, namely the SPT and the ST. Higher cost may hinder the applicability of our algorithm to large-scale topologies such as the Internet. Hence, to keep the communication cost as low as possible, the algorithm's search process is segmented into two different stages. The rationale is to put tighter limits on the node space by searching locally in the neighborhood (or vicinity) of the joining leaf node before searching globally. Indeed, the likelihood of finding a node of the MDT within a few hops distance from the joining leaf is high in large topologies (whose diameter is logarithmically proportional to its number of nodes) and this likelihood increases with the size of the MDT. Hence, we segment the search process by executing first a local search covering the leaf node's vicinity ball, and, if unsuccessful, by performing a global search over the remaining topology. By limiting the size (or order) of the vicinity ball taking into account the degree of the node it comprises, one ensures an optimal communication cost. For this purpose, a variable path budget π_b is used to limit the distance travelled by leaf initiated requests to prevent costly (in terms of communication) local search or global search. Additionally, as the most costly searches are resulting from the initial set of leaf nodes joining the multicast traffic source, each source constructs a domain (referred to as source ball). When a request reaches the boundary of that domain it is directly routed to the source.

III. COMPACT MULTICAST ROUTING ALGORITHM

This section describes the design of the proposed compact multicast routing algorithm. We first provide an overall description of the proposed algorithm. Then, we detail the local and global search phases used to discover the least cost branching path from the joining leaf node to the MDT. We also specify the design of the on-line and distributed construction algorithm underlying the incremental least cost branching path discovery process.

A. *Description*

The multicast distribution tree $T_{s,D}$ is constructed iteratively. At each step ω ($\omega = 1,2,..,|D|$) of the leaf-initiated construction, a randomly selected node u joins $T_{s,M}$, where $M \subseteq D$ corresponds to the current set of nodes part of the MDT at a given construction step. If node u is already part of $T_{s,M}$ ($u \in V_T$) then it is either a transit or a branching node of the MDT. Otherwise, node u is not part of $T_{s,M}$ ($u \in D \setminus V_T$) and it must search for the least cost branching path from node u to node v $\in T_{s,M}$. Among the set $P_{u,v}$ of possible paths p(u,v) from node $u \notin T_{s,M}$ to node $v \in T_{s,M}$, the least cost branching path p(u,v)* is defined as follows:

$$p(u,v)^* = \min\{c(u,v) \mid p(u,v) \in P_{u,v}\} \qquad (1)$$

In this equation, the cost c(u,v) of the path p(u,v) is defined as the sum of the cost c(u,w) of the edge (u,w), where w=succ(u) refers to the upstream neighbor node of u, and the cost c(w,v) of the path p(w,v). When each node along the least-cost branching path p(u,v)* from leaf node u to $v \in T_{s,M}$ determines its upstream neighbor node along that path, leaf node u can send to its selected upstream neighbor node a request message to join $T_{s,M}$. The join message is relayed along the selected least-cost branching path p(u,v)* until it reaches node $v \in T_{s,M}$. Once node u has joined $T_{s,M}$, $u \in V_T$, and the set M comprises node u, we proceed to the next step by randomly selecting a node $w \in D \setminus V_T$.

At the end of the iterative construction process, when all candidate leaf nodes have joined the MDT, i.e., M = D and $D \setminus V_T = \varnothing$, $T_{s,M} = T_{s,D}$. The routing table of each node $v \in T_{s,M}$ ($v \in V_T$) includes i) one routing table entry (stored in the multicast routing information base or MRIB) that indicates the upstream neighbor node to which the join message is sent for each source node s; this locally stored information enables performing Reverse Path Forwarding check so as to ensure loop-free forwarding of the incoming multicast packets, and ii) one multicast traffic routing entry (stored in the tree information base or TIB) to enable forwarding of incoming multicast traffic (generated from that source s) from its incoming port to a set of outgoing ports.

B. Least Cost Branching Path Discovery

Two types of messages are involved at each step of the least cost branching path discovery process, namely the request (type-R) messages flowing in the upstream direction towards the multicast source s, and the response (type-A) messages sent in the downstream direction towards the joining leaf node u.

- *Type-R message*: each message comprises the following information i) a sequence number $\{u_{id}, r_{id}\}$ to prevent duplication of messages, where u_{id} identifies the leaf node u and r_{id} identifies its request to join the multicast source s, ii) the leaf node u's timer value $\tau(u)$ that sets the waiting time at intermediates nodes before answering back to the downstream neighbor node, and iii) a path budget π, starting at leaf node u from $\pi(u) = \pi_{max}$, set at leaf node u. The π_{max} value is bound by the graph diameter (the length of the longest shortest path) for which approximation algorithms exist, as well as method for computing a lower and upper bounds [8]. Starting from leaf node u, the path budget $\pi(u)$ is decremented at each node v according to the travelled distance: each traversed edge accounts for a distance decrease of 1. When the type-R message reaches node v, if $\pi(v) = 0$, the latter does not further propagate the type-R message in order to keep the communication cost as low as possible; otherwise, the value $\pi(v)$ is decremented and passed to the neighboring nodes.

- *Type-A message*: sent in response to type-R messages, each type-A message comprises i) the cost $c(w,v)^*$ of the locally selected least cost path $p(w,v)^*$ from the local node w to v such that $v \in T_{s,M}$; a node $v \notin T_{s,M}$ generating a type-A message to its downstream neighbor nodes sets this cost to infinite, and ii) when $v \notin T_{s,M}$, the identifier of node v .

Both types of messages comprise a dedicated tag, called flag_e, which enables distinguishing between messages exchanged during the search phases. Both type-R and type-A messages are tagged as internal when setting flag_e=0 (if belonging to the local search procedure), and as external when flag_e=1, otherwise.

1) Local Search

This first stage consists in a limited search within a certain perimeter around the joining leaf node u. The contiguous set of nodes covered during this first stage is called the vicinity ball $B \subset V$. Each node $b \in B$ is therefore referred to as a vicinity node. The vicinity ball B of node u, B(u), is delimited by vicinity edge nodes, $b_v(u)$, i.e., nodes $v \in V$ at a given hop-count distance from node u.

At leaf node u, the path budget π carried in the type-R message is initialized by setting its value $\pi(u) = \pi_{max}$. If the degree of node u, $degree(u) \leq \alpha$ then $\pi(u) = 3$, if $\alpha \leq degree(u) < \beta$, $\alpha < \beta$, then $\pi(u) = 2$; otherwise $\pi(u) = 1$. Values α and β are small integer values determined a priori from the node degree distribution characterizing power law graphs. Starting from node u, where $\pi(u) = \pi_{max}$, the path budget π is decremented by 1 at each node v if $\pi(v) < \beta$; otherwise it is set to 1. This condition prevents propagation of the type-R message beyond adjacent nodes of high degree nodes. At node v, if the decremented $\pi(v)$ value reaches 0, node v does not further propagate the type-R message in order to keep the communication cost as low as possible while increasing the likelihood of finding a node $v \in T_{s,M}$. This procedure determines the maximum distance that type-R messages with flag_e=0 can traverse and determines the edge nodes of node's u vicinity ball B(u). Indeed, vicinity edge nodes $b_v(u)$ are the nodes for which the path budget π reaches 0.

When a given leaf node u decides to join the multicast source s, it sends a type-R message to all the direct upstream neighbor nodes of node u (referred to as succ(u)) to find the least cost branching path $p(u,v)^*$ to a node $v \in T_{s,M}$ ($v \in V_T$). At condition that succ(u) has not yet processed a type-R message with the same sequence number, succ(u) successively propagate the message following a split horizon until it reaches either a node $v \in T_{s,M}$ or a node $v \notin T_{s,M}$ and $\pi(v) = 0$. In the latter case, a vicinity edge node v is reached (node $v = b_v$) but no node belonging to $T_{s,M}$ can be found. Role of vicinity edge nodes is described in Section III.B.2.

At this point, node v replies to its downstream neighbor node(s) from which it has received the type-R message(s) with a type-A message. The type-A messages sent by node v in response to its downstream neighbor nodes $w = pred(v) \notin T_{s,M}$ are processed as follows. If node $v = b_v \notin T_{s,M}$, then the type-A message (issued by node v to its downstream neighbors) sets the branching path cost to infinite. If not, then the type-A message (issued by node v to its downstream neighbors) sets this cost to value 0 which indicates that node $v \in T_{s,M}$. Subsequently, each node $w \neq b_v$ $v = succ(w)$, computes the branching path costs $c(w,v)$ from itself to each node v by using (1), where either $v \in T_{s,M}$ or $v = b_v \notin T_{s,M}$. Node w then selects the least cost branching path $p(w,v)^*$ and sends the corresponding cost value $c(w,v)^*$ to its own downstream node(s) x such that $w = succ(x)$. Observe that each node w maintains no additional routing information besides the degree(w) entries required at each step of the execution. At waiting timer $\tau(u)$ expiration, if the set of type-A messages received by node u is empty or if the cost $c(succ(u),v)$ is set to infinite in all received type-A message, node u declares the multicast source s unreachable. Otherwise, leaf node u determines among its neighbor nodes succ(u) from which it received type-A messages, the upstream node succ(u*) along the least-cost branching path $p(u,v)^*$ ($= \min\{c(u,succ(u^*)) + c(succ(u^*),v) \mid p(u,v) \in P_{u,v}\}$) from node u to $v \in T_{s,M}$. Node u then further proceeds by sending a message to succ(u) to join $T_{s,M}$.

2) *Global Search*

This stage represents the search of the MDT's branching node outside the vicinity of the joining leaf node. This process is triggered by the leaf node u when the local search phase declares the multicast source s as unreachable in its vicinity ball B(u). The global search phase is triggered by the leaf node u and starts at each vicinity edge node $b_v(u)$. During this search phase, type-R and type-A messages tagged as external (i.e., flag_e=1).

In order to start a global search phase without restarting from the local neighborhood of the triggering node, the following procedures are considered:

- The first procedure enables external type-R messages reaching the vicinity edge nodes without traveling again the complete set of nodes inside its vicinity ball B(u). For this purpose, the leaf node u sends the external type-R messages directly to each of its vicinity edge nodes. Targeted forwarding of these messages from the leaf node u to each vicinity edge node b_v is possible because i) during the local search phase, the internal type-A messages (i.e., flag_e=0) received by the leaf node u include the identifier of the node b_v that initiates them, and ii) each vicinity nodes b ∈ B(u) keeps per vicinity edge node $b_v(u)$ a single active interface from which type-A message with infinite cost has been received (indicating that the neighbor node sits along the path from leaf node u to a given edge node b_v).
- The second prevents that a given node b ∈ B(u) receives back external type-R messages during the global search phase. For this purpose, vicinity edge node b_v filter incoming external type-R messages (i.e., flag_e=1). Remember that during the local search, internal type-A messages sent in response to the reception of type-R message (flag_e=0) are tagged with the flag_e=0. Interfaces sending such type-A message are removed from the list of interfaces for relaying type-R message (flag_e=1). The exception is for interfaces having received a type-R message (flag_e=1) with leaf node u as sender to enable edge vicinity nodes to send back the answer to node u once the global search completes for that node $b_v(u)$.

During the global search phase, the π(u) budget value is set at node u to a threshold equal to the graph diameter (length of the longest shortest path) and the waiting time τ(u) to a value that prevent waiting indefinitely. Moreover, upon reception of the type-R message (flag_e=1) from node u, each edge vicinity node $b_v(u)$ sets the maximum waiting timer $\tau(b_v(u))$ to τ(u) - 1. The subsequent search process proceeds as follows: assume that node $b_v(u)$ sends an external type-R message to each of its upstream neighbor nodes except to its downstream node (part of the vicinity ball of the node from which the message has been received). At waiting timer $\tau(b_v(u))$ expiration, if the set of type-A messages received by node $b_v(u)$ is empty or if the cost $c(succ(b_v(u)),v)$ is set to infinite in all received type-A message, node $b_v(u)$ declares the multicast source s as unreachable. Otherwise, node $b_v(u)$ determines among its neighbor nodes $succ(b_v(u))$ from which it received a type-A message, the upstream node $succ(v_b(u))^*$ along the least-cost branching path $p(v_b(u),v)^*$ to $T_{s,M}$, defined as:

$$p(b_v(u),v)^* = \min\{c(b_v(u),succ(b_v(u))^*) + c(succ(b_v(u))^*,v) \mid p(b_v(u),v) \in P_{b(u),v}\}. \quad (2)$$

Node $b_v(u)$ is ready to answer back to node u once either of the following condition is met: i) it receives the entire set of type-A messages from its upstream neighbor nodes before its waiting timer $\tau(b_v(u))$ expires or ii) the waiting timer $\tau(b_v(u))$ initiated after reception of the first type-R message (flag_e=1) from leaf node u expires. When one of these two conditions is met, node $b_v(u)$ selects the least cost branching path $p(b_v(u),v)^*$ and sends the corresponding cost value, $c(u,v)^*$ directly to the joining node u. At waiting timer $\tau(b_v(u))$ expiration, the set of type-A message received by node $b_v(u)$ is empty, the cost value $c(u,v)^*$ is set to infinite indicating that the multicast source s is unreachable. Hence, as soon as this search phase completes, each node $b_v(u)$ returns a unique type-A message (flag_e=1) directly to the leaf node u from which it initially received an external type-R message. Thus, contrary to the local search stage, no computation or selection is performed by nodes b ∈ B(u) along the path taken by the type-A messages (flag_e=1) sent towards the leaf node u. This relay path is determined by the incoming interface maintained by each node b ∈ B(u) upon reception of type-R message (e=1) from leaf node u. Note that the temporary records locally created during the local search phase are subsequently deleted by the node sending a type-A message (flag_e=0) that does not include an infinite cost to a vicinity edge node $b_v(u)$. The remaining records, locally created during the global search phase, are deleted by the node sending a type-A message (flag_e=1).

C. *Source Node Vicinity Ball*

As the most costly searches are resulting from the initial set of leaf nodes joining the MDT, each source constructs a source ball such that when a type-R message reaches the boundary of that domain it is directly routed to the source. This prevents searching at the neighborhood of the multicast traffic source. For this purpose, the multicast source node initiates a procedure that builds a ball around the size shall be at least as big as the average leaf node size. To be effective, this procedure shall construct a ball with i) size at least as large as the average size of leaf node's vicinity ball, and ii) radius computed from its outgoing ports shall be inversely proportional to the neighbor's node degree.

The procedure performs as follows: the source collects its neighbor's node degree to reach at a minimum size $x = x_{min}$ but up to a certain maximum diameter, d_{max}. At each collection step the following conditions are checked:

- If the value x reaches the optimal value of the ball size x_{opt} and diameter $d = d_{opt} \leq d_{max}$, then we get an optimal solution. The source ball construction stops.
- If $x > x_{opt}$ and the diameter d being reached is such that $d << d_{opt}$, then the source node s selects additional neighbor nodes so as to increase the diameter (up to value d_{max}) while limiting the increase above the value

x_{opt}. For this purpose, we define an epsilon ε of increase of x such that $x - \varepsilon \rightarrow x_{opt}$.

Epsilon ε is the number of adjacencies added from a given source node's neighbor t at distance d(s,t) to a set of adjacent neighbors at distance d(s,t) + 1 from s so as to reach a decent diameter while limiting the number of nodes in excess above x_{opt}. In practice, if $x > x_{opt}$, source node s selects neighbors located at a distance d(s,t) the nodes with smallest degree. Note that each node $b_v(s) \in B(s)$, the vicinity ball of the source node s, must now maintain an additional MRIB entry to relay type-R and type-A messages towards the source node s. Thus, in total 2.(#sourceBall_nodes-1) additional routing table entries are to be considered. Note however that if a leaf node $u = b_v(s) \in B(s)$, then that node u does not need any more to trigger any search procedure.

IV. Theoretical Performance Bounds

In this section, we provide the theoretical performance bounds of the proposed algorithm in terms of i) the stretch of the point-to-multipoint routing paths it produces, ii) the memory space required to store the resulting routing table entries, and iii) the total communication or messaging cost.

A. Stretch

Minimizing the tree-cost sequentially, namely, the total cost of the edges used during the algorithm while building the multicast tree of the various stages and minimizing the tree-cost globally leads to different stretch bounds. As we consider a dynamic join scenario, the former is considered. The stretch bound analysis involves three different cases:

- Consider a joining node u and $s \in B(u)$. Then the local search initiated by node u will find the least cost branching path if the path budget $\pi(u)$ in the type-R message initiated is sufficient to reach the source node s. It is obvious to see that if this condition is met, the resulting stretch increase is minimal.

- Consider a joining node u and $s \notin B(u)$. If $v \in T_{s,M}$ and $v \in B(u)$, then the local search process will find the actual least cost branching path if and only if there no other node $w \in T_{s,M}$ from the joining node u that can be found at shorter distance, i.e., $d(u,w) \geq d(u,v)$. Indeed, the distance limit set by the joining node on the local search process by means of the path budget $\pi(u)$ (see Section III), allows to reach a node $v \in B(u)$ before triggering a global search. Due to the degree bound set when decrementing the path budget $\pi(u)$, a node $v \in B(u)$ may be reached during the local search phase before reaching node $w \in T_{s,M}$ and $w \notin B(u)$ such that $d(u,w) < d(u,v)$. Henceforth, the stretch increase is bound by the fraction of such nodes conditioned by the joining node selection and the current number of nodes $\in T_{s,M}$. The stretch bound can thus be derived from the following formula:

$$\frac{1}{N}\sum_{i=1}^{N}\frac{d(u,v_i)}{\min d(u,w_i)} \qquad (3)$$

where, $\forall\, i \in M$, $|M| = N$, $\exists\, v_i, w_i \in V$: $\min d(u,w_i) < d(u,v_i)$ with a probability $P(v_i \in T_{s,M} \mid v_i \in B(u))\,.\,P(w_i \in T_{s,M} \mid w_i \notin B(u)) > 0$.

- Consider a joining node u and $s \notin B(u)$. If $v \in T_{s,M}$ and $v \notin B(u)$, then the global search process will find the least cost branching path if the path budget $\pi(u)$ in the type-R message initiated is sufficient to reach the source node s. It is obvious to see that if this condition is met, the resulting stretch increase is minimal.

B. Storage

Each node $v \in T_{s,M}$ stores in its local routing table at most one MRIB entry and at most one TIB entry whose size is proportional to the local tree out-degree k (as this entry indicates the outgoing ports for the incoming multicast traffic). Assuming a minimum port encoding proportional to log(k), the storage size per node $v \in T_{s,M}$ is O(k log(k)) bits. Nodes $b_v(s) \in B(s)$ require an additionally storage capacity for the MRIB enabling to relay type-R and type-A messages.

C. Communication

Each join event as initiated by a node u_i results in a communication cost $C(u_i)$, i.e., the number of messages exchanged for node u_i to join a node $v \in T_{s,M}$, that is given by the following formula:

$$C(u_i) = 2m'X + 2m'Y + 2(m - m')Z \qquad (4)$$

In this equation, m and m' are respectively the number of edges in the vicinity ball B(u) of node u and the total number of edges |E|. X is the probability that at least one node $v \in T_{s,M}$ is comprised in the ball B(u) of the selected node $u \in V$, Y is the probability that none of the tree nodes $v \in T_{s,M}$ are comprised in the ball B(u) of the selected node $u \in V$, and Z is the probability that all nodes $v \in V$ part of the tree $T_{s,M}$ are not in the ball of the selected node u (all tree nodes lie outside of the ball B(u) of the selected node u). The total communication cost, i.e., the cost to build the entire MDT is then determined by the sum of the individual communication costs $C(u_i)$ of the i = 1,…,|N| leaves composing the tree. In [7], we demonstrate that the derivative of (4) with respect to the number of nodes in the vicinity of node u, |B(u)|, provides the value of the ball size that minimizes the communication cost while corresponding to the order of the largest connected sub-graph of diameter d_{max} that can be constructed.

V. Performance Analysis

The performances of the proposed compact multicast routing algorithm are analyzed by means of simulation on large scale topologies generated by GLP [9] and Internet CAIDA maps. The GLP evolutive topology model, which relies on generalized linear preferential attachment, produces power law graphs that are representative of the Internet Autonomous System (AS) topology (one node models an AS), in particular, in terms of clustering coefficient. The properties of these topologies are summarized in Table I.

The execution scenario considers the construction of point-to-multipoint routing paths for leaf node set of increasing size from 500 to 4000 nodes (selected randomly) with increment of 500 nodes. Each execution is performed 10 times by considering 10 different multicast sources. We compare the performance of our algorithm to the Shortest-Path Tree (SPT) and the Steiner Tree (ST) algorithms.

- The SPT algorithm provides the reference for the communication cost. It is constructed from a loop-avoidance path-vector routing algorithm carrying the identifier of the multicast source s and the routing path to reach that source. Each node keeps thus a routing table (RT) entry per neighbor node (to exchange messages) and a RT entry per path to the source s.
- The ST algorithm provides the reference in terms of stretch. In order to obtain the near optimal solution for the ST, we consider a ST-Integer Linear Programming formulation. For this purpose, we adapted the formulation provided in [10] for bi-directional graphs. The communication cost for the ST measures at each step of the MDT construction the number of messages initiated by nodes part of the MDT. These messages contain the minimal information for remote nodes not (yet) belonging to the MDT to join it. Using this information, each node knows how to reach the closest node of the MDT. Thus, although the ST is computed centrally, the communication cost accounts for the total number of messages exchanged during the MDT building process as a dynamic scenario would perform.

TABLE I: TOPOLOGY PROPERTIES

Topology Properties		
Topology Property	*GLP*	*CAIDA*
Nodes - Links	32618 - 146816	32000 - 120436
Avg - Max Node Degree	4,50 - 2520	7,53 - 1165

A. Stretch

This section details the simulation results obtained for the multiplicative stretch defined as the cost ratio between the point-to-multipoint routing paths (underlying the MDT) produced by the proposed scheme and the minimum Steiner Tree. We also compare the cost ratio between the point-to-multipoint routing path produced by the SPT and the minimum Steiner Tree.

1) GLP Topology

As shown in Figure 1, the multiplicative stretch for the proposed algorithm is slightly higher than 1 for the GLP topology. As the leaf node set increases from 500 to 4000, its trend curve decreases from 1.09 (maximum value reached for 500 leaf nodes) to 1.05 (minimum value reached for 4000 leaf nodes). Compared to the SPT stretch, our algorithm maintains an average gain of 4% along the different group sizes.

2) CAIDA Map

As shown in Figure 2, the multiplicative stretch for the proposed algorithm is slightly higher than 1 for the GLP topology. As the leaf node set increases from 500 to 4000, its trend curve decreases from 1.08 (maximum value reached for 500 leaf nodes) to 1.03 (minimum value reached for 4000 leaf nodes). Compared to the SPT stretch, our algorithm maintains a maximum deterioration of 4% for sets of 500 leaf nodes; this deterioration becomes negligible as the size of the leaf node sets increases.

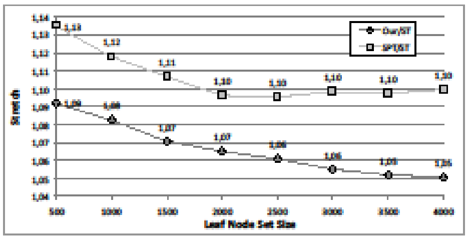


Figure 1. Stretch as a function of Leaf Node Set Size

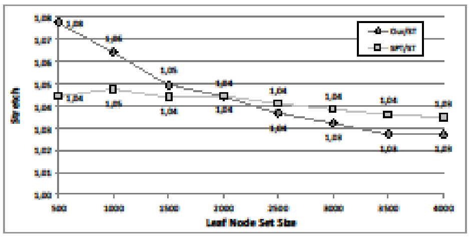


Figure 2. Stretch as a function of Leaf Node Set Size

B. Storage

This section details the simulation results obtained for the memory capacity required to store the routing tables entries (underlying the MDT) produced by proposed scheme. It also determines the relative gain we obtain in terms of the ratio between the total number of RT entries produced by our algorithm and the total number of RT produced by the reference algorithms. This ratio provides an indication of the achievable reduction in terms of the memory capacity required to store the routing table entries produced by these algorithms.

1) GLP Topology

From Table II, we can observe that the proposed algorithm (Our) produces significantly less RT entries that the ST and SPT reference algorithms. The highest number of RT entries is obtained for a set of 4000 leaf nodes: 10154 RT entries. This value is 4,10 times smaller than the number of RT entries produced by the ST algorithm (41643 RT entries) and 27,92 times smaller than the number of the RT entries produced by the SPT algorithm (283477 RT entries).

Figure 3 illustrates the relative gain expressed in terms of the ratio between the total number of RT entries produced by the ST and the SPT references and our algorithm. An increasing gain can be observed as the size of the leaf node set decreases from 4,10 (leaf set of 4000 nodes) to 20,34 (leaf set of 500 nodes) compared to the ST algorithm and from 27,92 (leaf set of 4000 nodes) to 166,08 (leaf set of 500 nodes) compared to the SPT algorithm.

TABLE II: NUMBER OF RT ENTRIES FOR THE SPT, ST, AND OUR ALGORITHM WITH RESPECT TO THE LEAF NODE SET SIZE.

	Routing Scheme		
Leaf Node Set Size	SPT	ST	Our
500	274555	33483	1646
1000	275927	34733	2986
1500	277261	35965	4272
2000	278561	37191	5554
2500	279827	38349	6756
3000	281045	39443	7874
3500	282265	40559	9022
4000	283477	41643	10154

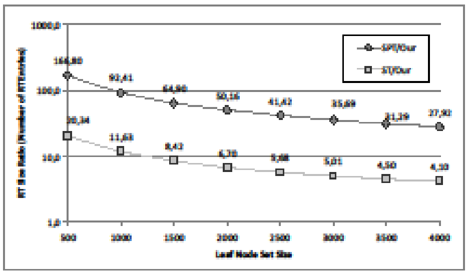


Figure 3. RT Size Ratio as a function of Leaf Node Set Size

2) CAIDA Map

From Table III, the proposed algorithm (Our) produces significantly less routing table entries that the ST and SPT reference algorithms. The highest number of RT entries is obtained for set of 4000 leaf nodes: 13169 RT entries. This value is 3,21 times smaller than the number of RT entries produced by the ST algorithm (42277 RT entries) and 14,38 times smaller than the number of the RT entries produced by the SPT algorithm (189431 RT entries).

TABLE III: NUMBER OF RT ENTRIES FOR THE SPT, ST, AND OUR ALGORITHM WITH RESPECT TO THE LEAF NODE SET SIZE.

	Routing Scheme		
Leaf Node Set Size	SPT	ST	Our
500	180993	34111	4919
1000	182339	35391	6237
1500	183609	36609	7471
2000	184825	37779	8653
2500	186009	38935	9807
3000	187151	40047	10921
3500	188325	41199	12059
4000	189431	42277	13169

Figure 4 illustrates the relative gain expressed in terms of the ratio between the total number of RT entries produced by the ST and SPT references and our algorithm. An increasing gain can be observed as the size of the leaf node set decreases from 3,21 (leaf set of 4000 nodes) to 6,93 (leaf set of 500 nodes) compared to the ST algorithm and from 14,38 (leaf set of 4000 nodes) to 36,79 (leaf set of 500 nodes) compared to the SPT algorithm. Interestingly, the obtained gain values for the CAIDA map are smaller than those obtained for the GLP topology. This difference can be explained resulting from the difference in tree-depth: 6 (leaf set of 500 nodes) to 9 (leaf set of 4000 nodes) for the CAIDA map vs 8 (leaf set of 500 nodes) to 11 (leaf set of 4000 nodes) for the GLP topology.

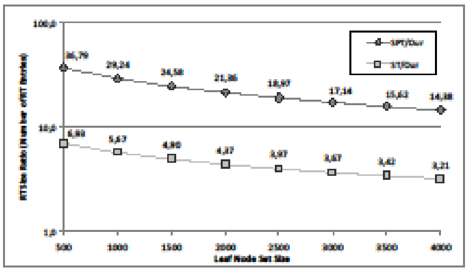


Figure 4. RT Size Ratio as a function of Leaf Node Set Size

C. *Communication Cost*

The communication cost is a critical metric to determine the applicability of the proposed compact routing algorithm to large scale topologies comprising as in the present work, i.e., 32k nodes. The two-stage search procedure and the source node vicinity ball construction, both presented in Section III, play an important role in mitigating the communication cost.

1) GLP Topology

As depicted in Figure 5, the communication cost ratio for the proposed algorithm is relatively high compared to the SPT even if much lower than the communication cost implied by the ST (not represented in this figure). Indeed, the communication cost ratio increases from 2,69 (leaf set of 500 nodes) to 8,17 (leaf set of 4000 nodes). This observation can be explained by the presence of high degree nodes (nodes that have a degree of the order to 100 or even higher) in power law graphs. However, as computed this communication cost does not take into account for the evolution of the routing topology. This evolution impacts multicast routing algorithms such as the SPT that are strongly dependent on non-local unicast routing information compared to the proposed algorithm. Moreover, as shown in Figure 5, the communication cost of the proposed algorithm compared to the SPT communication cost, decreases as the number of nodes composing the leaf node set increases. This trend leads us to expect that a saturation level can be reached around a communication cost ratio not higher than 10 to 15 as the size of the lead node set continues to grow. It is worth mentioning that the memory and the capacity required to process communication messages are relatively limited.

2) CAIDA Map

The same trend can be observed for the CAIDA Map where the communication cost ratio between our scheme and the SPT algorithm increases from 7,88 (leaf set of 500 nodes) to 13,77 (leaf set of 4000 nodes). The difference observed between the CAIDA map and the GLP topology can be explained from the following observation the tree-depth differs by a unit (3 vs 4). This difference induces a relatively higher cost of the SPT when running over the GLP topology.

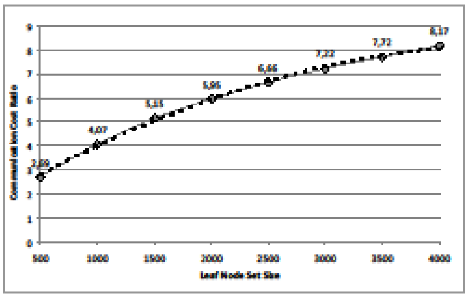


Figure 5. Communication Cost Ratio as a function of Leaf Node Set Size

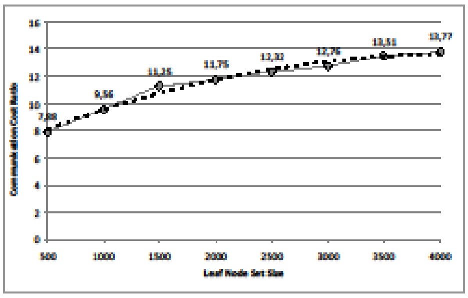


Figure 6. Communication Cost Ratio as a function of Leaf Node Set Size

D. *Comparison with the Abraham Scheme*

Here below, the performance in terms of the stretch of the point-to-multipoint routing paths produced and the memory space required by the proposed algorithm and by the Abraham routing scheme as specified in [5] (for dynamic join only events) are compared.

1) Stretch

For the scheme allowing only dynamic join events, the MDT cost is given by Lemma7 of [5]. The authors determine that the proposed dynamic multicast algorithm is $O(\min\{\log n, \log \Delta\}. \log n)$ competitive compared to the cost of the optimal algorithm – Steiner Tree. In this formula, the factor Δ is the aspect ratio defined as the ratio between max $d(u,v)$ and min $d(u,v)$, for any $u, v \in V$. Considering an aspect ratio Δ of 6 and a network of 32k nodes the stretch is about 3.5. Thus the stretch upper bound of the point-to-multipoint routing path produced by the Abraham scheme, even if universal (applicable to any graph), is about 3 times higher than the one produced by our scheme.

2) Storage

Following the description of the Abraham scheme provided in Section II.B, the storage requirement is given by the memory space that includes i) the tree routing information $\mu(T,v)$ stored by each node v, for all trees in its own SPLabel(v) leading to a total storage of $O(\log^3 n.\log\Delta/\log \log n)$ bits, ii) for each $i \in I$ and $T \in TC_{k,2i}(G)$, the center node $c(T(v))$ of each node $v \in T$ that stores the labels of all nodes contained in the ball $B(v,2^i)$ leading to a total storage over all radii of $O(kn^{1+1/k} \log \Delta)$ bits; in addition, each node v stores $O(\log \Delta)$ labels of size $\tilde{O}(kn^{1/k})$ each leading to a total memory consumption of $\tilde{O}(kn^{1+1/k})$ bits. The resulting memory storage requires about 700kbits for a tree comprising 4000 leaf nodes. For the same leaf set size, our routing scheme requires about 1250kbits.

VI. Conclusion

This paper introduces the first known name-independent compact multicast routing algorithm enabling the leaf-initiated, distributed and dynamic construction of MDT. The performance obtained shows substantial gain in terms of the number of RT entries compared to the ST (minimum factor of 3,21 for sets of 4000 leaf nodes, i.e., 12,5% of the topology size) and the memory space required to store them. The stretch deterioration compared to the ST algorithms ranges between 8% and 3% (for multicast group size of 500 to 4000, respectively); thus, decreasing with increasing group sizes. The proposed two-phase search process -local search first covering the leaf's node vicinity, and if unsuccessful, a global search over the remaining topology- combined with the vicinity ball construction at the source node enables to keep the communication cost of the proposed algorithm within reasonable bounds compared to the reference SPT scheme and sub-linearly proportional to the size of the leaf node set. Future work will determine if these promising performance results can still be verified for dynamic sequences of node join and node leave events and non-stationary topologies.

References

[1] D.Peleg and E.Upfall, "A trade-off between space and efficiency forrouting tables," J. ACM, vol.36, no.3, pp.510–530, Jul.1989.

[2] B.Awerbuch, et.al., "Compact distributed data structures for adaptive routing," Proc. 21st annual ACM STOC'89, Seattle (WA), USA, May.1989.

[3] M.Thorup, and U.Zwick, "Compact routing schemes," Proc. 13th Annual ACM SPAA'01, Heraklion, Crete, Greece, pp.1–10, Jul.2001.

[4] I.Abraham, et al., "Compact name-independent routing with minimum stretch," ACM Trans. Alg., vol.4, no.3, art.37, Jun.2008.

[5] I.Abraham, D.Malkhi, and D.Ratajczak, "Compact multicast routing," Proc. 23rd Int. Symp. DISC'09, Elche, Spain, pp.364–378, Sep.2009.

[6] B.Fenner, et.al., "Protocol Independent Multicast - Sparse Mode (PIM-SM)," Internet Engineering Task Force (IETF), RFC 4601, Aug.2006.

[7] P.Pedroso, D.Papadimitriou, and D.Careglio, "A name-independent compact multicast routing algorithm", available as Technical Report, UPC-DAC-RR-CBA-2011-15, Jul. 2011.

[8] C.Magnien, et al., "Fast computation of empirically tight bounds for the diameter of massive graphs," J. Exper. Alg., vol.13, art.10, Feb.2009.

[9] T.Bu, and D.Towsley, "On distinguishing between Internet power law topology generators," Proc. IEEE Infocom'02, New York (NJ), USA, Jun.2002.

[10] Sage's Graph Library. Available at http://www.sagemath.org/