An Algebraic Theory of Dynamic Network Routing

João Luís Sobrinho, *Member, IEEE*

***Abstract*—We develop a non-classic algebraic theory for the purpose of investigating the convergence properties of dynamic routing protocols. The algebraic theory can be regarded as a generalization of shortest-path routing, where the new concept of free cycle generalizes that of a positive-length cycle. A primary result then states that routing protocols always converge, though not necessarily onto optimal paths, in networks where all cycles are free. Monotonicity and isotonicity are two algebraic properties that strengthen convergence results. Monotonicity implies protocol convergence in every network, and isotonicity assures convergence onto optimal paths.**

A great many applications arise as particular instances of the algebraic theory. In intra-domain routing, we show that routing protocols can be made to converge to shortest and widest paths, for example, but that the composite metric of Internet Gateway Routing Protocol (IGRP) does not lead to optimal paths. The more interesting applications, however, relate to inter-domain routing and its Border Gateway Protocol (BGP), where the algebraic framework provides a mathematical template for the specification, design, and verification of routing policies. We formulate existing guidelines for inter-domain routing in algebraic terms, propose new guidelines contemplating backup relationships between domains, and derive a sufficient condition for signaling correctness of internal-BGP.

***Index Terms*—Algebra, convergence, inter-domain routing, intra-domain routing, routing protocols.**

I. Introduction

NON-CLASSIC algebra has made headway in many branches of electrical engineering and computer science, from coding and cryptography to compiler design and networking, unifying seemingly unrelated concepts and establishing fundamental results. Can it also shed light into distributed network routing, especially as witnessed in the Internet protocols? We answer affirmatively by defining suitable algebraic structures and exploring their properties.

Packets in the Internet are forwarded by routers as a function of their destinations, regardless of their origins. The forwarding table at each router is kept up to date by dynamic routing protocols that react to network failures and additions. Routing is administratively divided in intra-domain and inter-domain, each with its own set of goals and protocols. A domain is a collection of routers under the administrative and operational control of a single entity. Intra-domain routing is, in general, performance-oriented: the entity administering the domain aims at the best use of its internal network resources. Routing Information Protocol (RIP) [1], [2], Interior Gateway Routing Protocol (IGRP) [3], and Open Shortest Path First (OSPF) [4], [5] are examples of intra-domain routing protocols, the first and second of distance-vector type, and the third of link-state type [6]. By contrast, inter-domain routing is policy-oriented: the various domains forward packets toward the destinations as a function of local policies that reflect the commercial relationships established between them. The Border Gateway Protocol (BGP) [7], [8] is currently the only protocol for inter-domain routing and is a path-vector protocol [6].

The purpose of the algebraic theory herein presented is to establish fundamental results on the convergence of routing protocols, and see them applied in a variety of different environments. The convergence results are formulated in terms of path-vector protocols, but they bear as well to distance-vector protocols. Link-state protocols call for a more specific, less general, algebraic theory, which is presented in [9].

An algebra for routing comprises a set of labels, a set of signatures, and a set of weights. Each network link has a label and each network path has a signature. There is an operation to obtain the signature of a path from the labels of its constituent links, and a function mapping signatures to weights. Ultimately, each path will have a weight, and these weights are ordered such that any set of paths with the same origin and destination can be compared: the lower the weight of a path the more preferred the path is. For example, if labels, signatures, and weights are real numbers, composed by standard addition and ordered by the standard less-than-or-equal relation, the resulting instance of the algebra represents shortest-path routing. In light of this, the algebraic theory can be seen as providing a generalization of shortest-path routing.

A central concept is that of free cycle. In a free cycle, for any collection of paths, each starting at a different node of the cycle, at least one of these paths weighs less than the path that starts at the same node, proceeds to the node's neighbor around the cycle, and continues with the path that starts at the neighbor. Intuitively, if a cycle is free, then, given any destination in the network, at least one of its nodes forwards packets to the destination out of the cycle, instead of around the cycle, thus preventing packets from being trapped in a loop. In the instance of the algebra representing shortest-path routing, the free cycles are exactly the positive-length cycles, so that the former cycles generalize the latter. Moreover, the generalization carries over to the role those cycles play on the convergence of routing protocols. In shortest-path routing, it is known that path-vector protocols converge if all network cycles have positive length [10]. For the broader algebraic theory, we show that path-vector protocols converge if all network cycles are free.

Some algebras are enriched by properties that intertwine its elements, allowing for stronger statements about protocol convergence to emerge, and permitting a computationally easy

Manuscript received February 16, 2004; revised October 8, 2004, and October 19, 2004; approved by IEEE/ACM Transactions on Networking Editor J. Crowcroft.

The author is with the Instituto de Telecomunicações, Instituto Superior Técnico, 1049-001 Lisboa, Portugal (e-mail: joao.sobrinho@lx.it.pt).

Digital Object Identifier 10.1109/TNET.2005.857111

characterization of free cycles. Two such properties are considered here: monotonicity and isotonicity. Monotonicity implies that the weight of a path does not decrease when prefixed by a link. If the algebra is monotone, then every network can be made free, thereby insuring convergence of path-vector protocols. Isotonicity implies that the order relationship between the weights of any two paths with the same origin is preserved when both are prefixed by the same link. If the algebra is isotone, then the paths onto which path-vector protocols converge are optimal.

Many applications can be drawn from the general theory. In performance-oriented environments, we conclude, for example, that path-vector protocols can be used to make packets travel over shortest or widest paths, but that the composite metric of IGRP does not make them travel over optimal paths [11]. The more interesting applications of the algebraic framework, however, are to policy-oriented routing and BGP. We formulate the guidelines of Gao and Rexford [12] in algebraic terms, both at the domain level and at the router level, present new guidelines that exploit backup relationships between Internet domains, and provide a sufficient condition for signaling correctness of Internal-BGP (IBGP) [13]–[17]. A couple of different applications can be found in our previous work [18].

We address related work in Section II. The algebra is presented in Section III. An example path-vector protocol is described in Section IV. Freeness and its relation to protocol convergence are examined in Section V. Monotonicity and isotonicity are dissected in Sections VI and VII, respectively. Applications of the algebra in performance-oriented environments and policy-oriented environments are discussed in Sections VIII and IX, respectively.

II. Related Work

The monographs by Carré [19] and by Gondran and Minoux [20], [21] cover great many algebraic structures and supporting sequential algorithms to solve optimization problems defined over networks, and they served as inspiration for our work. However, the problems addressed here are different from those enunciated in [19]–[21], leading to different algebraic structures. On the one hand, we are interest in convergence properties of routing protocols, which are distributed rather than sequential algorithms. On the other hand, optimality of the converged solutions is too strong a requirement in most scenarios, notably in those that pertain to inter-domain routing. Freeness and monotonicity are two properties that care for the convergence of routing protocols without concern for the optimality of the converged solutions.

Griffin *et al.* [22], [23] were the first to come up with a comprehensive model to study convergence of path-vector protocols. In their model, each network node is explicitly assigned an ordered set of permitted paths, through which a given destination can be reached. A sufficient condition for convergence is related to a property that intertwines the various sets of permitted paths. The algebraic model presented here is positioned at a higher level of abstraction, bringing two main advantages. On the one hand, an algebra provides a semantic context for the specification and design of routing strategies. On the other hand, properties of an algebra and of networks labeled with its elements relate directly to the convergence properties of path-vector protocols, and they can typically be verified at low computational complexity.

III. Algebra for Routing

A. Basic Definitions

A network is modeled as a directed graph. The presence of link (u,v) in a network means that packets can flow from u to v, and that signaling routing messages may be sent in the opposite direction, from v to u. Link (u,v) has node u for *head* and node v for *tail*; we say that node v is an *out-neighbor* of node u, and that node u is an *in-neighbor* of node v.

A *walk* in a network is a sequence of nodes $u_n u_{n-1} \cdots u_1 u_0$ such that (u_i, u_{i-1}) is a network link for $1 \leq i \leq n$. Note that we choose to index the nodes of a walk backward, from last to first. A *path* is a walk where all nodes are distinct, and a *cycle* is a walk where all nodes are distinct except for the first and last. The *order* of a walk is the number of links it contains. A path of order zero is called *trivial*. Given walks P and Q where the last node of P is the first node of Q, we denote their concatenation by $P \circ Q$. Specifically, if $P = u_n u_{n-1} \cdots u_1 u_0$ and $Q = v_m v_{m-1} \cdots v_1 v_0$, with $u_0 = v_m$, then $P \circ Q = u_n u_{n-1} \cdots u_1 u_0 v_{m-1} \cdots v_1 v_0$. For the special case where uv is a path with only two nodes, we say that $uv \circ Q$ is the *extension* of walk Q to node u, or that $uv \circ Q$ is the extension of walk Q by link (u,v).

B. Algebra

An algebra for routing is an ordered septet $(W, \preceq, L, \Sigma, \phi, \oplus, f)$. It comprises:

- a set of *weights* W;
- a set of *labels* L;
- a set of *signatures* Σ;
- a total order $\preceq$ on W;
- a binary operation $\oplus$ that maps pairs with a label and a signature into a signature;
- a function f that maps signatures into weights;
- the special signature ϕ.

The relation $\prec$ on W is defined such that $a \prec b$ if $a \preceq b$ and $a \neq b$; the relation $\succeq$ is defined such that $a \succeq b$ if $b \preceq a$; and the relation $\succ$ is defined such that $a \succ b$ if $a \succeq b$ and $a \neq b$. Every algebra for routing has at least the following two properties.

Absorption For all $l \in L$, $l \oplus \phi = \phi$.

Maximality For all $\alpha \in \Sigma - \{\phi\}$, $f(\alpha) \prec f(\phi)$.

An algebra for routing is finite if Σ is finite, in which case the set of labels and the set of weights can also be taken finite. The set of labels can be assumed finite because every label defines a mapping from signatures to signatures via operation $\oplus$ and there are $|\Sigma|^{|\Sigma|}$ such distinct mappings;[1] the set of weights can be assumed finite because the range of f is finite.

Although an algebra for routing has an existence of its own, regardless of networks or routing protocols, its elements and properties have been defined with these concepts in mind, to ultimately arrive at conclusions related to the convergence of

[1] $|S|$ denotes the cardinality of set S.

routing protocols. The links of a network are assigned labels from the set L, with $l(u,v)$ denoting the label of link (u,v). The walks of the network are assigned signatures from the set Σ, with $s(P)$ denoting the signature of walk P. The signature of walk P is obtained from the labels of its constituent links through composition with operation $\oplus$. If P is the trivial path consisting only of node u, then $s(u)$ is an intrinsic property of node u; otherwise, P can be written as $uv \circ P'$, for some nodes u and v and walk P', and $s(P) = l(u,v) \oplus s(P')$. The special signature ϕ is reserved for *unusable* walks, which are those that cannot be used for packet transport. Any walk with signature different from ϕ is said to be *usable*. The absorption property implies that the extension of an unusable walk by a link produces another unusable walk. The mapping f from signatures to the totally ordered set of weights W results in an assignment of weights to walks, with the weight of walk P being given by $f(s(P))$. It thus establishes a ranking among walks. Informally, the lower the weight of a walk, according to the order $\preceq$, the "better." The maximality property implies that any usable walk is "better" than an unusable one. An *optimal* path from u to d is a usable path from u to d of minimum weight, that is, whose weight is less than or equal, according to the order $\preceq$, to that of any walk from u to d. Hence, an optimal path is "better" or "equally good" as any walk from u to d.

The most familiar example of an algebra for routing is the one that leads to shortest-path routing, henceforth called the shortest-path algebra. In this case, labels, signatures, and weights represent lengths: labels are real numbers; signatures are indistinguishable from weights, being real numbers adjoined with the special element $+\infty$; the signature of a walk is obtained by adding the labels of its constituent links; and weights are compared with the less-than-or-equal order. The shortest-path algebra is thus the septet $(\mathbb{R} \cup \{+\infty\}, \leq, \mathbb{R}, \mathbb{R} \cup \{+\infty\}, +\infty, +, \mathrm{id})$,[2] where id denotes the identity function on $\mathbb{R} \cup \{+\infty\}$. Absorption follows easily from $l + (+\infty) = +\infty$, for all $l \in \mathbb{R}$, and maximality follows from $\mathrm{id}(s) = s < +\infty = \mathrm{id}(+\infty)$, for all $s \in \mathbb{R}$. An optimal path in this algebra is a standard shortest path.

C. Optimal and Local-Optimal In-Trees

An *in-tree* is a subgraph of a network satisfying the following three clauses:

- it has only one node, called the *root*, without out-neighbors;
- all its other nodes have one and only one out-neighbor;
- there is an in-tree path from every one of its nodes to the root.

An in-tree does not have to be spanning: not all network nodes need to be part of it. Fig. 1 shows an in-tree rooted at node 0. In-trees are the graph structures one expects to find when forwarding packets based only on their destination addresses, as is usually the case in the Internet.

An *optimal in-tree* rooted at d is an in-tree rooted at d which, in addition, satisfies the following two clauses:

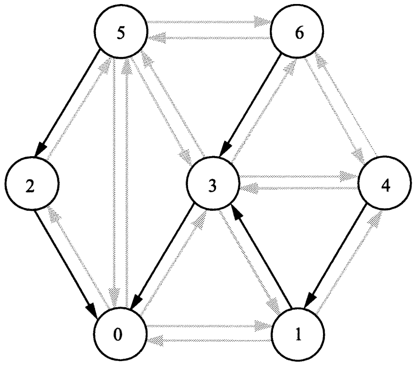

Fig. 1. The dark links represent an in-tree rooted at node 0.

- node u belongs to the in-tree if and only if there is a usable walk in the network from u to d;
- the in-tree path from u to d is optimal in the network.

A *local-optimal in-tree* rooted at d is an in-tree rooted at d which, in addition, satisfies the following two clauses:

- node u belongs to the in-tree if and only if it has an out-neighbor v in the in-tree such that $uv \circ P'$ is a usable walk in the network, where P' is the in-tree path from v to d;
- the in-tree path from u to d has weight less than or equal to that of any walk in the network of the form $uv \circ P'$ with v an out-neighbor of u in the in-tree and P' the in-tree path from v to d.

In the shortest-path algebra, every local-optimal in-tree is optimal as well, but this assertion does not carry over to all other algebras. In Sections VIII and IX, we will see examples of algebras for which a local-optimal in-tree is not necessarily optimal. The converse assertion always holds, however: whatever the algebra, an optimal in-tree is also local-optimal.

IV. Path-Vector Protocol

A. Description

We consider a collection of nodes wanting to exchange packets by making use of the links of a network. Links fail and are added in the course of time, presumably at a rate much lower than that of packet transmissions. A routing protocol is a distributed algorithm that maintains the forwarding tables, one per node, that collectively guide packets toward their destinations. In destination-based packet forwarding, the forwarding tables contain correspondences between destinations and out-neighbors that should bring packets closer to those destinations.

A path-vector protocol is a routing protocol whereby the basic unit of signaling information kept at the nodes and exchanged between them is either a pair of the form (P, α), with P a usable path to a destination and α its signature, or else the pair $(none, \phi)$, where $none$ stands for the absence of a path to reach a destination. We call such pairs *couplets*. The terminology we have used for paths carries over to couplets, so that, for instance, the weight of couplet (P, α) is $f(\alpha)$, and we say that couplet (P, α) is usable if α is not ϕ, in which case its origin and destination are those of path P. Note that, because of the dynamics

[2] $\mathbb{R}$ denotes the set of real numbers, $\mathbb{R}^+$ denotes the set of positive reals, $\mathbb{R}_0^+$ denotes the set of non-negative reals, $\mathbb{N}_0$ denotes the set of non-negative integers, and $\mathbb{N}$ denotes the set of positive integers.

of the system, the couplets known by the nodes and being announced in the network at a given time may be stale. For example, when a node receives a signaling routing message reporting usable couplet (P, α), path P may no longer exist in the network—some link of P might have failed—or its signature may no longer be α—some link of P might have failed and have been repaired with a different label.

Fixing a destination, at any given time each node knows of a couplet to reach the destination by going through each one of its out-neighbors. The node chooses one of those couplets and installs in the forwarding table the correspondence between the destination and the out-neighbor associated with the chosen couplet. The chosen couplet is always one of minimum weight from among those known by the node. If there is more than one couplet of minimum weight, the node deterministically chooses one of them. We assume that the relative preference given to equal-weight couplets having the same origin and the same destination totally orders those couplets. The strict partial order $\lhd$ on usable couplets captures these relative preferences: $(P, \alpha) \lhd (Q, \beta)$ if both couplets have the same origin and the same destination, and either (P, α) weighs less than (Q, β) or they have the same weight but (P, α) is preferred to (Q, β) at their common origin.

Algorithm 1 presents representative path-vector protocol code for node u. This code is executed atomically when node u receives couplet (P, α) from its out-neighbor v' pertaining to destination node d. The pair of variables $(ptab_u[v, d], stab_u[v, d])$ holds the couplet currently known by node u to reach node d through out-neighbor v, and the pair of variables $(path_u[d], sign_u[d])$ holds the couplet currently chosen at node u to reach node d. Algorithm 1 states that once node u receives couplet (P, α), it first updates the pair of variables $(ptab_u[v', d], stab_u[v', d])$ to reflect a new couplet to reach d through v'. If u is not a node of P and $l(u, v') \oplus \alpha$ differs from ϕ, then $uv' \circ P$ is a usable path to d through v', which is stored in variable $ptab_u[v', d]$, with signature $l(u, v') \oplus \alpha$ being stored in variable $stab_u[v', d]$. Otherwise, if either u is a node of P or $l(u, v') \oplus \alpha$ equals ϕ, then there is no usable path to d through $v', none$ is stored in $ptab_u[v', d]$, and ϕ is stored in $stab_u[v', d]$. After computing the new couplet to reach d through v', node u chooses the most preferred, minimum weight couplet from among the couplets stored in the pairs of variables $(ptab_u[v, d], stab_u[v, d])$, with v an out-neighbor of u, and copies it to the pair of variables $(path_u[d], sign_u[d])$. If the chosen couplet is usable, the associated out-neighbor is installed in the forwarding table entry corresponding to destination d. Otherwise, if the chosen couplet is unusable, destination d is declared unreachable in the forwarding table. Finally, the chosen couplet is advertised to all in-neighbors of node u but only if it has changed with the reception of the signaling routing message. Similar code exists to deal with the failure or addition of a link. We assume that for each link (u, v) in the network, the signaling routing messages in transit from v to u are delivered at u is the order sent by v and they are only lost if the link fails.

Algorithm 1 Protocol code when node u receives couplet (P, α) from out-neighbor v' pertaining to destination d.

```
if $u$ is not a node of $P$ and $l(u, v') \oplus \alpha$ is not $\phi$ then
    $ptab_u[v', d] := uv' \circ P$
    $stab_u[v', d] := l(u, v') \oplus \alpha$
else
    $ptab_u[v', d] := none$
    $stab_u[v', d] := \phi$
if there is $v$ such that $stab_u[v, d]$ is not $\phi$ then
    let $v^*$ be such that, for every out-neighbor $v \neq v^*$,
        $(ptab_u[v^*, d], stab_u[v^*, d]) \lhd (ptab_u[v, d], stab_u[v, d])$
    $path_u[d] := ptab_u[v^*, d]$
    $sign_u[d] := stab_u[v^*, d]$
else
    $path_u[d] := none$
    $sign_u[d] := \phi$
if $(path_u[d], sign_u[d])$ has changed then
    for all $v$ in-neighbor of $u$ do
        send couplet $(path_u[d], sign_u[d])$ to $v$
```

Some variations of Algorithm 1 can be found in implementations. For example, in the last two lines of code, if node u can determine that node v is already part of path $path_u[d]$, or that $l(v, u) \oplus sign_u[d]$ equals ϕ, it may send couplet $(none, \phi)$ to in-neighbor v, instead of couplet $(path_u[d], sign_u[d])$. Also, the signature of a couplet may be omitted if it can be inferred from the enumeration of the nodes that make up the path and the label of the link joining the recipient to the sender of the couplet. These variations do not alter our main conclusions.

B. Specification

The specification of every path-vector protocol contains at least a liveness requirement and a basic safety requirement. The liveness requirement imposes convergence of the protocol. That is, if no more links fail or are added from a certain time onwards, then there is a future instant of time when no more signaling routing messages are to be found in transit in the network. The basic safety requirement imposes that the paths onto which the protocol has converged form in-trees, one for each destination. Other safety requirements may be imposed depending on the application. A typical requirement found in performance-oriented routing is the optimality requirement, which states that the in-trees onto which the protocol converges should be optimal in-trees.

V. Network Freeness and Protocol Convergence

In this section we introduce the concepts of free cycle and free network and relate them to the convergence of path-vector protocols. Define the function S that maps walk P and signature α into signature $S(P, \alpha)$ as follows: if P is a trivial path, then $S(P, \alpha) = \alpha$; otherwise, P can be written as $P = uv \circ P'$, for some nodes u and v and walk P', and $S(P, \alpha) = l(u, v) \oplus S(P', \alpha)$. Therefore, $S(P, \alpha)$ would be the signature of walk P if α were the intrinsic signature of the trivial path composed of its last node alone. In particular, $s(P) = S(P, s(d))$, with d the last node of P.

Freeness Cycle $u_n u_{n-1} \cdots u_1 u_0$, with $u_n = u_0$, is free if for every $\alpha_0, \alpha_1, \ldots, \alpha_{n-1}, \alpha_n \in \Sigma - \{\phi\}$, with $\alpha_0 = \alpha_n$, there is an index $i, 1 \leq i \leq n$, such that $f(\alpha_i) \prec f(S(u_i u_{i-1}, \alpha_{i-1}))$.

Freeness of a cycle implies that given any collection of walks, each walk starting at a different node of the cycle, at least one of these walks will weigh less than the walk that starts at the same node, proceeds to this node's out-neighbor in the cycle,

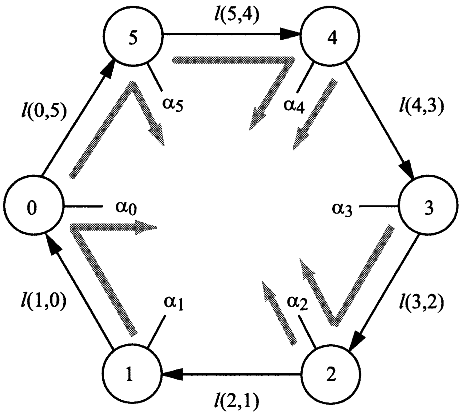

Fig. 2. Cycle 0 5 4 3 2 1 0 and a particular assignment of signatures to its nodes such that $f(\alpha_1) \succeq f(S(1\,0,\alpha_0)), f(\alpha_2) \prec f(S(2\,1,\alpha_1)), f(\alpha_3) \succeq f(S(3\;2,\alpha_2)), f(\alpha_4) \prec f(S(4\;3,\alpha_3)), f(\alpha_5) \succeq f(S(5\;4,\alpha_4))$, and $f(\alpha_0) \succeq f(S(0\;5,\alpha_5))$.

and then continues with the walk that starts at the out-neighbor. Intuitively, if a cycle is free, then, given any destination in the network, at least one of its nodes forward packets to the destination out of the cycle, instead of around the cycle, thus preventing packets from being trapped in a loop. Fig. 2 shows a cycle and an assignment of signatures for which the signatures assigned to nodes 2 and 4 weigh less than the signatures obtained by composing the labels of the links joining these nodes to their out-neighbors with the signatures assigned to the out-neighbors.

Further insight into the concept of free cycle is obtained by uncovering its meaning in the shortest-path algebra. As the next proposition shows, a free cycle in the shortest-path algebra is exactly a positive-length cycle, so that free cycles can be regarded as a generalization of positive-length cycles.

Proposition 1: A cycle is free in the shortest-path algebra if and only if it has positive length.

Proof: Recall that the shortest-path algebra is the ordered septet $(\mathbb{R}\cup\{+\infty\}, \leq, \mathbb{R}, \mathbb{R}\cup\{+\infty\}, +\infty, +, \mathrm{id})$. Suppose first that cycle $u_n u_{n-1} \cdots u_1 u_0$, with $u_n = u_0$, is free, and let $\alpha_0 = \alpha_n = 0$ and $\alpha_i = l(u_i, u_{i-1}) + \cdots + l(u_1, u_0)$ for $1 \leq i < n$. From these definitions, we get $\mathrm{id}(\alpha_i) = \alpha_i = l(u_i, u_{i-1}) + \alpha_{i-1} = \mathrm{id}(S(u_i u_{i-1}, \alpha_{i-1}))$, for $1 \leq i < n$. Therefore, for the cycle to be free, it must be the case that $\mathrm{id}(\alpha_n) < \mathrm{id}(S(u_n u_{n-1}, \alpha_{n-1}))$. Developing this inequality yields

$$\begin{aligned} 0 &= \mathrm{id}(\alpha_n) \\ &< \mathrm{id}(S(u_n u_{n-1}, \alpha_{n-1})) \\ &= l(u_n, u_{n-1}) + \alpha_{n-1} \\ &= l(u_n, u_{n-1}) + l(u_{n-1}, u_{n-2}) + \cdots + l(u_1, u_0) \end{aligned}$$

meaning that cycle $u_n u_{n-1} \cdots u_1 u_0$ has positive length. To prove the converse, suppose that cycle $u_n u_{n-1} \cdots u_1 u_0$, with $u_n = u_0$, is not free. Then, there are $\alpha_0, \alpha_1, \ldots, \alpha_{n-1}, \alpha_n \in \mathbb{R}$, with $\alpha_0 = \alpha_n$, such that

$$\begin{aligned} \alpha_i &= \mathrm{id}(\alpha_i) \\ &\geq \mathrm{id}(S(u_i u_{i-1}, \alpha_{i-1})) \\ &= l(u_i, u_{i-1}) + \alpha_{i-1} \end{aligned}$$

for $1 \leq i \leq n$. Adding these n inequalities yields

$$0 \geq l(u_n, u_{n-1}) + l(u_{n-1}, u_{n-2}) + \cdots + l(u_1, u_0)$$

meaning that the length of cycle $u_n u_{n-1} \cdots u_1 u_0$ is not positive. ■

We have implicitly used several properties of the real numbers in deducing the equivalence between free cycles and positive-length cycles, in the shortest-path algebra. In general, and for the finite case, one would need to consider $(|\Sigma| - 1)^n$ signature combinations to check whether or not a cycle of order n is free. This computational complexity is reduced if the algebra has more structure as discussed further in Sections VI and VII.

For the shortest-path algebra, it is well-known that path-vector protocols converge to shortest-paths if all cycles in the network have positive-length [10]. A generalization of this result is given in the following theorem, where we say that a network is free if all its cycles are free.

Theorem 1: In a free network, the path-vector protocol converges to local-optimal in-trees.

Theorem 1 is the conjunction of a liveness property and a safety property. The liveness property corresponds to the liveness requirement; it states that if the network onto which the system settles down is free, then the path-vector protocol converges. The safety property tells us a bit more than the basic safety requirement; it states not only that once the path-vector protocol has converged it has converged onto in-trees, but also that these in-trees are local-optimal. A semi-formal proof of Theorem 1 supported on temporal logic [24], [25] is presented in the Appendix.

VI. Monotonicity

Monotonicity An algebra for routing is monotone if for all $l \in L$ and $\alpha \in \Sigma$, $f(\alpha) \preceq f(l \oplus \alpha)$.

In a network, monotonicity implies that the weight of a walk does not decrease when it is extended by a new link. Checking whether or not a finite algebra is monotone entails $|L| \times |\Sigma|$ binary operations and that same number of binary comparisons, assuming that the signatures are sorted in increasing order of weights.

For every $w \in W - \{f(\phi)\}$, define the set

$$L_w = \{l \in L | \text{there is}\, \alpha \in \Sigma \,\text{such that}\, w = f(\alpha) = f(l \oplus \alpha)\}.$$

If label l belongs to L_w, then there is a walk that maintains its weight w when it is extended by a link with label l. The sets $L_w, w \in W - \{f(\phi)\}$, can be constructed as we check for monotonicity, and they allow for an easy characterization of free cycles.

Theorem 2: In a monotonic algebra, cycle C is free if and only if for every weight $w \in W - \{f(\phi)\}$ there is a link in the cycle whose label does not belong to L_w.

Proof: We first show the forward implication. Write cycle C as $u_n u_{n-1} \cdots u_1 u_0$, with $u_n = u_0$, and suppose there is $w \in W - \{f(\phi)\}$ such that $l(u_i, u_{i-1}) \in L_w$ for all $i, 1 \leq i \leq n$. Then, for every link $(u_i, u_{i-1}), 1 \leq i \leq n$, in C there is $\alpha_{i-1} \in \Sigma - \{\phi\}$ such that $w = f(\alpha_{i-1}) = f(l(u_i, u_{i-1}) \oplus \alpha_{i-1}) = f(S(u_i u_{i-1}, \alpha_{i-1}))$. In addition, let $\alpha_n = \alpha_0$. Therefore, $f(\alpha_i) = w = f(S(u_i u_{i-1}, \alpha_{i-1}))$, for $1 \leq i \leq n$, showing that cycle C is not free.

To prove the reverse implication, suppose that cycle C is not free. Then, there are $\alpha_0, \alpha_1, \ldots, \alpha_{n-1}, \alpha_n \in \Sigma - \{\phi\}$, with $\alpha_0 = \alpha_n$, such that $f(\alpha_i) \succeq f(S(u_i u_{i-1}, \alpha_{i-1}))$ for all $i, 1 \leq i \leq n$. Because the algebra is monotone, we have

$$\begin{aligned} f(\alpha_i) &\succeq f(S(u_i u_{i-1}, \alpha_{i-1})) \\ &= f(l(u_i, u_{i-1}) \oplus \alpha_{i-1}) \\ &\succeq f(\alpha_{i-1}) \end{aligned}$$

for $1 \leq i \leq n$. This set of inequalities implies $f(\alpha_i) = f(l(u_i, u_{i-1}) \oplus \alpha_{i-1}) = f(\alpha_{i-1})$, for $1 \leq i \leq n$. Letting w denote the weight common to all signatures α_i, we have that $l(u_i, u_{i-1}) \in L_w$. ■

Theorem 2 is perhaps easier to comprehend and apply in the following form: cycle C is non-free if and only if all its links' labels belong to a common set $L_w, w \in W - \{f(\phi)\}$. Therefore, we can verify whether or not a cycle C is free by testing membership of its links' labels in a common set L_w, where each such set has a maximum of $|L|$ elements. A special case of monotonicity is strict monotonicity. An algebra for routing is strictly monotone if for all $l \in L$ and $\alpha \in \Sigma - \{\phi\}, f(\alpha) \prec f(l \oplus \alpha)$. In strict monotonic algebras, L_w is empty whatever $w \in W - \{f(\phi)\}$, every network is free, and the path-vector protocol always converges. A statement of this fact, arrived at with a different formulation, appears in [26]. For the case in which the algebra is monotone, but not necessarily strict monotone, we have the following theorem.

Theorem 3: If the algebra is monotone, then the path-vector protocol can be made to converge to local-optimal in-trees whatever the network.

Proof: Informally, the idea is to break the non-free cycles by appending to the weight of a walk its order, effectively forcing nodes to always prefer couplets of lowest order among couplets of equal weight. Non-free cycles can only be broken this way if the algebra is monotone.

From the original algebra $(W, \preceq, L, \Sigma, \phi, \oplus, f)$, we define a primed algebra $(W', \preceq', L, \Sigma', \phi, \oplus', f')$, such that: $W' = (W - \{f(\phi)\}) \times \mathbb{N}_0 \cup \{f(\phi)\}$; $\Sigma' = (\Sigma - \{\phi\}) \times \mathbb{N}_0 \cup \{\phi\}$; $(\alpha, n) \preceq' (\beta, m)$ if $\alpha \prec \beta$, or if $\alpha = \beta$ and $n < m$; $l \oplus' (\alpha, n) = (l \oplus \alpha, n+1)$; and $f'((\alpha, n)) = (f(\alpha), n)$. The intrinsic primed signature of node u is $(s(u), 0)$. Any couplet that weighs less than another in the original algebra also weighs less in the primed algebra. However, among couplets with the same weight in the original algebra, the primed algebra gives preference to the ones of lowest order. The primed algebra is strictly monotone implying that every network is free. From Theorem 1, the path-vector protocol converges. ■

The converse to Theorem 3 also holds: if the algebra is not monotone, then we can find a network for which path-vector protocols do not converge. If the algebra is not monotone, then there are $l \in L$ and $\alpha \in \Sigma$ such that $f(\alpha) \succ f(l \oplus \alpha)$. In particular, $\alpha \neq \phi$. In the network of Fig. 3, node 0 is the destination, and $\alpha = s(P_1) = s(P_2)$. Suppose that signaling routing messages incur a delay of exactly one unit of time traveling either from 1 to 2 or from 2 to 1. At time zero, nodes 1 and 2 have just chosen couplets (P_1, α) and (P_2, α) to reach node 0, respectively, and advertised these choices to each other. After one unit of time has elapsed, node 1 learns of couplet (P_2, α) and, because $f(l \oplus \alpha) \prec f(\alpha)$, it changes its chosen couplet to $(1\,2 \circ P_2, l \oplus \alpha)$; ditto for node 2, which changes its chosen couplet to reach 0 to $(2\,1 \circ P_1, l \oplus \alpha)$. After one more unit of time has elapsed, node 1 learns that node 2 has chosen couplet $(2\,1 \circ P_1, l \oplus \alpha)$ to reach 0. Since path $2\,1 \circ P_1$ contains node 1, it is not an option for node 1: node 1 reverts its couplet choice to (P_1, α). Similarly, node 2 reverts its couplet choice to (P_2, α). We are back at the initial conditions, the described sequence of events repeats itself, and the protocol never converges.

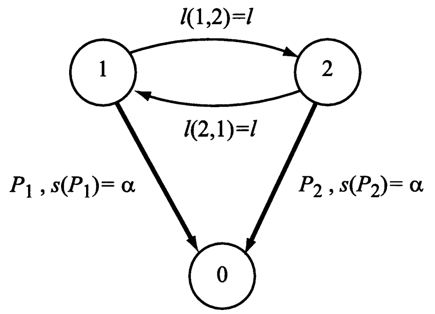

Fig. 3. Thick lines represent paths and thin lines represent links. Suppose that $f(\alpha) \succ f(l \oplus \alpha)$. Then, paths $1\,2 \circ P_2$ and $2\,1 \circ P_1$ weigh less than paths P_1 and P_2, respectively. If signaling routing messages are exchanged synchronously, then the path-vector protocol never converges.

The shortest-path algebra is not monotone, because if l is negative, then $\mathrm{id}(s) = s > l + s = \mathrm{id}(l + s)$ for all $s \in \mathbb{R}$. However, redefining the shortest-path algebra so that the set of labels is $\mathbb{R}_0^+$, rather than $\mathbb{R}$, makes it monotone. In this case, $L_w = \{0\}$ for every $w \in \mathbb{R}$. From Theorem 2, we conclude that a cycle is non-free if and only if all its links have label 0, or equivalently, if and only if the cycle has zero length. Because the labels are constrained to be non-negative, the previous statement is further equivalent to that of Proposition 1: a cycle is free if and only if it has positive length. Invoking Theorem 3, the path-vector protocol can be made to converge even in networks with cycles of zero length, if each node always prefers paths with the minimum number of links among paths of the same length. The shortest-path algebra would be strict monotone if the set of labels were to be further constrained to $\mathbb{R}^+$.

VII. Isotonicity

Isotonicity An algebra for routing is isotone if for all $l \in L$ and $\alpha, \beta \in \Sigma, f(\alpha) \preceq f(\beta) \Rightarrow f(l \oplus \alpha) \preceq f(l \oplus \beta)$.

In a network, isotonicity implies that the weight relationship between two walks with the same origin is preserved when both are extended by a common link. Checking whether or not a finite algebra is isotone entails $|L| \times |\Sigma|$ binary operations and $|L| \times |\Sigma| \times (|\Sigma| - 1)/2$ binary comparisons, assuming that signatures are sorted in increasing order of weights. As was the case with monotonicity, isotonicity allows an easy, albeit different, characterization of free cycles.

Theorem 4: In an isotonic algebra, cycle C is free if and only if for every signature $\alpha \in \Sigma - \{\phi\}$ we have that $f(\alpha) \prec f(S(C, \alpha))$.

Proof: Let cycle C be written as $u_n u_{n-1} \cdots u_1 u_0$, with $u_n = u_0$. We start with the forward implication, for which isotonicity is not needed. Suppose that C is free and that we are given $\alpha \in \Sigma - \{\phi\}$. Define the α_i inductively as follows: $\alpha_0 = \alpha_n = \alpha$; and $\alpha_i = l(u_i, u_{i-1}) \oplus \alpha_{i-1}$ for $1 \leq i < n$. Therefore,

$S(u_iu_{i-1}\cdots u_1u_0,\alpha) = S(u_iu_{i-1},\alpha_{i-1})$ for $1 \leq i < n$. If $\alpha_i = \phi$ for some $0 \leq i < n$, then $f(\alpha) \prec f(\phi) = f(S(C,\alpha))$. Otherwise, because $f(\alpha_i) = f(S(u_iu_{i-1},\alpha_{i-1}))$, for $1 \leq i < n$, and cycle C is free, we must have

$$\begin{aligned} f(\alpha) &= f(\alpha_n) \\ &\prec f(S(u_nu_{n-1},\alpha_{n-1})) \\ &= f(S(C,\alpha)) \end{aligned}$$

concluding the forward implication.

For the reverse implication, suppose that C is not free. Then, there are $\alpha_0,\alpha_1,\ldots,\alpha_{n-1},\alpha_n \in \Sigma - \{\phi\}$, with $\alpha_0 = \alpha_n$, such that the set of inequalities $f(\alpha_i) \succeq f(S(u_iu_{i-1},\alpha_{i-1}))$ holds for all $i, 1 \leq i \leq n$. We show by induction that $f(\alpha_i) \succeq f(S(u_iu_{i-1}\cdots u_1u_0,\alpha_0))$ for $1 \leq i \leq n$. The base case is $f(\alpha_1) \succeq f(S(u_1u_0,\alpha_0))$ which follows directly from the fact that C is not free. For the induction step, assume that $f(\alpha_{i-1}) \succeq f(S(u_{i-1}u_{i-2}\cdots u_1u_0,\alpha_0))$. From isotonicity, we obtain

$$\begin{aligned} &f(S(u_iu_{i-1},\alpha_{i-1})) \\ &\quad = f(l(u_i,u_{i-1}) \oplus \alpha_{i-1}) \\ &\quad \succeq f(l(u_i,u_{i-1}) \oplus S(u_{i-1}u_{i-2}\cdots u_1u_0,\alpha_0)) \\ &\quad = f(S(u_iu_{i-1}\cdots u_1u_0,\alpha_0)). \end{aligned}$$

Because C is not free, $f(\alpha_i) \succeq f(S(u_iu_{i-1},\alpha_{i-1}))$, so that, using as well the previous inequality, we get $f(\alpha_i) \succeq f(S(u_iu_{i-1}\cdots u_1u_0,\alpha_0))$, which completes the induction step. Now, taking $i = n$ and defining α to equal $\alpha_0 = \alpha_n$, yields $f(\alpha) = f(\alpha_n) \succeq f(S(u_nu_{n-1}\cdots u_1u_0,\alpha)) = f(S(C,\alpha))$, which is what we wanted to prove. ■

In a finite algebra, we only need to consider the $|\Sigma| - 1$ signatures of set $\Sigma - \{\phi\}$ to decide whether or not cycle C is free. Isotonicity also has a consequence on the characteristics of the local-optimal in-trees onto which the path-vector protocol converges: these in-trees are optimal.

Theorem 5: If the algebra is isotone, then every local-optimal in-tree is an optimal in-tree.

Proof: Suppose we are given a network with a local-optimal in-tree rooted at destination d. Let u be a network node with a usable walk to d, and let $u_nu_{n-1}\cdots u_1u_0$ be any such walk, with $u_n = u$ and $u_0 = d$. We will show by induction that node u_i belongs to the in-tree, and that the in-tree path P_i from u_i to u_0 weighs less than, or has the same weight as, walk $u_iu_{i-1}\cdots u_1u_0$, for $0 \leq i \leq n$. Then, the case $i = n$ yields the desired optimality of P_n.

The base case is trivial, since $P_0 = u_0$. For the induction step, assume that the in-tree path P_{i-1} from u_{i-1} to u_0 weighs less than, or has the same weight as, walk $u_{i-1}u_{i-2}\cdots u_1u_0$, that is, $f(s(P_{i-1})) \preceq f(s(u_{i-1}u_{i-2}\cdots u_1u_0))$, for $1 \leq i \leq n$. Applying isotonicity, we get

$$\begin{aligned} f(s(u_iu_{i-1} \circ P_{i-1})) &= f(l(u_i,u_{i-1}) \oplus s(P_{i-1})) \\ &\preceq f(l(u_i,u_{i-1}) \oplus s(u_{i-1}u_{i-2}\cdots u_1u_0)) \\ &= f(s(u_iu_{i-1}\cdots u_1u_0)). \end{aligned}$$

In particular, walk $u_iu_{i-1} \circ P_{i-1}$ is usable. Hence, u_i belongs to the in-tree, and because the in-tree is a local-optimal one, we have

$$f(s(P_i)) \preceq f(s(u_iu_{i-1} \circ P_{i-1})).$$

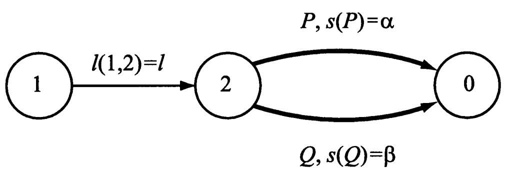

Fig. 4. Thick lines represent paths and thin lines represent links. Suppose that $f(\alpha) \preceq f(\beta)$, but $f(l \oplus \alpha) \succ f(l \oplus \beta)$. Then, the local-optimal in-tree rooted at 0 that contains path P, and not path Q, is not an optimal in-tree rooted at 0.

Combining the two inequalities, completes the desired induction proof. ■

Combining Theorems 1 and 5, we conclude that if the network is free and the algebra is isotone, then a path-vector protocol converges to optimal paths. The converse to Theorem 5 also holds: if the algebra is not isotone, then we can find a network for which a local-optimal in-tree is not an optimal in-tree. If the algebra is not isotone, then there are $l \in L$ and $\alpha,\beta \in \Sigma$ such that $f(\alpha) \preceq f(\beta)$ but $f(l \oplus \alpha) \succ f(l \oplus \beta)$. In Fig. 4, node 0 is the destination, path P has signature α, path Q has signature β, and link $(1,2)$ has label l. The in-tree that contains path P to the disadvantage of path Q is a local-optimal in-tree because $f(s(P)) = f(\alpha) \preceq f(\beta) = f(s(Q))$. As a consequence, the path in the local-optimal in-tree from 1 to 0 is path $1\,2 \circ P$, if any. However, that is not an optimal path from 1 to 0, since $f(s(1\,2 \circ P)) = f(l \oplus \alpha) \succ f(l \oplus \beta) = f(s(1\,2 \circ Q))$.

The shortest-path algebra is isotone, because if $\mathrm{id}(s) = s \leq t = \mathrm{id}(t)$, then $\mathrm{id}(l+s) = l+s \leq l+t = \mathrm{id}(l+t)$ for all $l \in \mathbb{R}$ and $s,t \in \mathbb{R} \cup \{+\infty\}$. Theorem 4 applied to the shortest-path algebra confirms what we already knew from Proposition 1: a cycle is free if and only if it has positive length.

From Theorem 5, we further conclude that every in-tree of local-shortest paths is an in-tree of shortest paths.

VIII. Applications to Performance-Oriented Routing

A. Standard Optimal-Path Routing

We have already introduced the shortest-path algebra $(\mathbb{R} \cup \{+\infty\}, \leq, \mathbb{R}, \mathbb{R} \cup \{+\infty\}, +\infty, +, \mathrm{id})$, and shown that it isotone but not monotone, and that the free cycles are the positive length cycles. For a diverse example, take the algebra that leads to widest-path routing. A widest path is a path of maximum capacity (width), where the capacity of a path equals that of its link of least capacity. The widest-path algebra is the ordered septet $(\mathbb{R}_0^+ \cup \{+\infty\}, \geq, \mathbb{R}^+ \cup \{+\infty\}, \mathbb{R}_0^+ \cup \{+\infty\}, 0, \min, \mathrm{id})$. This algebra is both monotone and isotone. Referring back to Section VI, we have $L_w = \{l \in \mathbb{R}^+ \cup \{+\infty\} | l \geq w\}$ for every $w \in \mathbb{R}^+ \cup \{+\infty\}$. Therefore, from Theorem 2, all cycles are non-free. Notwithstanding, path-vector protocols may still converge, as stated in Theorem 3. For convergence, it suffices to have each node prefer a couplet of lowest order among couplets of equal capacity. Other examples of optimal-path algebras can be found in [9].

B. Composite Metric of IGRP

The composite metric of IGRP [3] provides an example of an algebra that is monotone but not isotone, implying convergence of path-vector protocols, but not to optimal paths, against what one would expect to find in an intra-domain performance-oriented environment [11]. In its most basic form, the composite metric of IGRP can be described by an algebra with $L = \mathbb{R}^+ \times \mathbb{R}^+, \Sigma = L \cup \{\epsilon, \phi\}, W = \mathbb{R}_0^+ \cup \{+\infty\}$. The first component of a label represents length and the second represents capacity. Accordingly, $(d_1, b_1) \oplus \epsilon = (d_1, b_1)$, and $(d_1, b_1) \oplus (d_2, b_2) = (d_1 + d_2, \min(b_1, b_2))$ for $(d_2, b_2) \in \Sigma - \{\epsilon, \phi\}$. The order $\preceq$ is $\leq$, and the function f is given by

$$f(\epsilon) = 0, \quad f((d, b)) = d + \frac{k}{b}, \quad f(\phi) = +\infty$$

where k is a positive constant. It is easy to verify that the algebra is monotone. The failure of isotonicity can be exemplified with the inequalities $f((2, k)) = 3 < 5 = f((1, k/4))$, and $f((1, k/4) \oplus (2, k)) = f((3, k/4)) = 7 > 6 = f((2, k/4)) = f((1, k/4) \oplus (1, k/4))$. Because the first component of every label is positive, all networks are free.

IX. Applications to Policy-Oriented Routing

A. Primer on Inter-Domain Routing and BGP

In this section, we present sufficient information on inter-domain routing and BGP to contextualize the ensuing applications. Currently, the commercial relationships between Internet domains, also called Autonomous Systems (ASes), can be classified into customer-provider, peer-peer, and backup [12], [27]–[30]. A customer pays to its providers for connectivity to the Internet whereas any two peers agree to exchange traffic between their customers free of charge. Backup relationships maintain Internet connectivity in the event of link failures. The policies configured at the routers of an AS reflect the commercial relationships the AS has established with its neighboring ASes, and they consist of import/export and preference rules for routes. A route is the basic unit of information kept at BGP routers and exchanged between them. Common import/export rules state that [12], [27]:

- an AS does not export to a provider or peer routes that it learned from other providers and other peers;
- an AS can export to its customers any route it knows of.

The preference rules suggested in Guideline A of Gao and Rexford [12] state that routes learned from customers should be preferred to routes learned from either providers or peers, leaving ASes latitude to assign relative preferences among customer routes, and among peer and provider routes. The import/export and preference rules are realized with recourse to the BGP attributes associated with every route. The basic BGP attributes are LOCAL-PREF, AS-PATH, and MED [7]. LOCAL-PREF is a degree of preference locally assigned to a route; AS-PATH is the, possibly inflated, sequence of ASes traversed by a route; and MED discriminates among several links joining neighboring ASes. MED brings its own set of problems to routing which are not addressed here [31].

BGP can also be used to distribute routing information inside an AS. To distinguish BGP sessions established between two routers in different ASes from BGP sessions established between two routers in the same AS, the former are called external BGP (EBGP) sessions and the latter are called internal BGP (IBGP) sessions. The most basic scenario has each pair of routers in an AS holding an IBGP session, leading to a fully connected mesh of IBGP sessions. The import/export rules determine that a router does not export to other routers in the same AS routes learned via IBGP. An alternative to the fully connected topology is provided by route reflection [13]–[17]. In the simplest form of this strategy—the one we consider here—the routers inside an AS are partitioned into clusters. Each cluster contains one route reflector and a number of clients. IBGP sessions are established between every pair of route reflectors, and between a route reflector and every one of its clients. They may also be established between two clients in the same cluster. The import/export rules only permit the following exports [13]:

- a route learned by a route reflector from another is exported to all its clients;
- a route learned by a route reflector from one of its clients is exported to all its other clients and to all route reflectors;
- a route learned by a router via an EBGP session is exported through all the router's IBGP sessions.

Whether or not route reflection is used, a router with more than one route at the same level of external preference to reach a given destination—as defined by the LOCAL-PREF, AS-PATH, and MED attributes—chooses only one of these routes according to the following internal preference rules, taken in order [7]:

- routes learned via EBGP sessions are preferred to routes learned via IBGP sessions;
- routes with a shorter Internal Gateway Protocol (IGP) distance to the border router that announced the route into the AS are preferred;
- routes announced into the AS by border routers with lower identifiers are preferred.

B. Interdomain Routing at the as Level

We formulate Guideline A of Gao and Rexford [12] in algebraic terms, assuming that each network node represents an AS. We have $L = \{c, r, p\}, \Sigma = L \cup \{\epsilon, \phi\}$, and $W = \{0, 1, 2, +\infty\}$. The order $\preceq$ is $\leq$. Links joining providers to customers are called customer links, and have label c; links joining customers to providers are called provider links, and have label p; and links joining peers to other peers are called peer links, and have label r. We call *primary paths* to the usable paths obtained with the guidelines of this section. Primary paths are subdivided according to their signatures into four classes: trivial paths, comprised of a single node, have signature ϵ; customer paths, whose first link is a customer link, have signature c; provider paths, whose first link is a provider link, have signature p; and peer paths, whose first link is a peer link, have signature r.

The binary operation $\oplus$ is given in the next chart, where the first operand, a label, appears in the first column and the second operand, a signature, appears in the first row.

		signature			
	$\oplus$	ϵ	c	r	p
label	c	c	c	ϕ	ϕ
	r	r	r	ϕ	ϕ
	p	p	p	p	p

For example, $c \oplus r = \phi$ means that a peer path cannot be extended by a customer link. In other words, an AS does not export

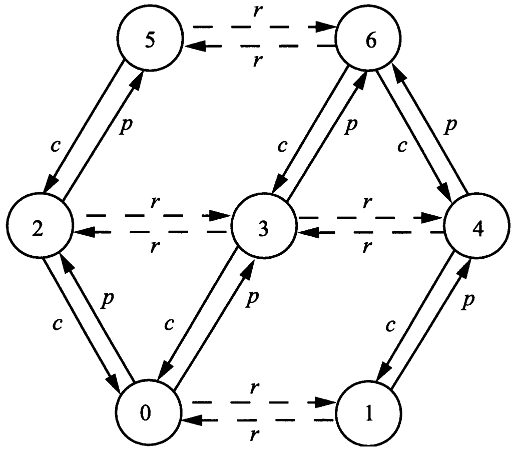

Fig. 5. Network with customer-provider and peer-peer relationships. Labels are taken from the set $\{c, r, p\}$, where c, r, and p, identify customer, peer, and provider links, respectively. Peer links are represented with dashed lines as a visualization aid.

to a provider a route learned from a peer. From the definition of operation $\oplus$, we deduce that any primary path is of the form $P \circ R \circ C$, where path P contains only provider links, path R is either a trivial path or a path formed by a single peer link, and path C contains only customer links. Any of the paths P, R, and C can be a trivial path. Fig. 5 depicts a network where links have labels taken from set L. Node 5 is a provider of node 2, and consequently, node 2 is a customer of node 5. Nodes 2 and 3 are peers. Link $(5, 2)$ is a customer link; link $(2, 5)$ is a provider link; and links $(2, 3)$ and $(3, 2)$ are peer links. Path 5 2 0 is a customer path; path 3 6 5 2 0 is a provider path; and path 3 2 0 is a peer path. Paths 5 2 3 0 and 2 0 3, for example, are not primary paths. The function f is given by

$$f(\epsilon) = 0, \quad f(c) = 1, \quad f(r) = f(p) = 2, \quad f(\phi) = +\infty.$$

The inequality $f(c) = 1 < 2 = f(r) = f(p)$ means that a node always prefers a customer path to either a peer path or a provider path.

The algebra is both monotone and isotone, so that a path-vector protocol can always be made to converge, and when it does, it converges to optimal paths, although that was not a requirement in the first place. Theorem 2 can be used to identify the free networks associated with this algebra. Scanning the pairs label-signature, we obtain: L_0 is the empty set, since $0 = f(\epsilon) \prec f(l \oplus \epsilon)$ for every $l \in L; L_1 = \{c\}$, since $1 = f(c) = f(c \oplus c) \prec f(p \oplus c) = f(r \oplus c)$; and $L_2 = \{p\}$, since $2 = f(r) = f(p \oplus r) \prec f(c \oplus r) = f(r \oplus r)$ and $2 = f(p) = f(p \oplus p) \prec f(c \oplus p) = f(r \oplus p)$. Hence, a cycle is non-free if either: i) all its links have label c or ii) all its links have label p: a free network is a network without such cycles. This is equivalent to the hierarchical assumption of [12] which states that: i) the subgraph of the network induced by the customer links alone and ii) the subgraph of the network induced by the provider links alone should be acyclic. In terms of the relationships established between Internet domains, a free Internet is a network where no domain is a provider of one of its direct or indirect providers. To guarantee convergence of the path-vector protocol without constraining the network, it suffices to have each domain break ties within paths of the same class—customer, provider, or peer—with the order of the path.

C. Interdomain Routing at the Router Level

The previous model does not anticipate the possibility of having two routers in the same AS choosing different paths to reach a given destination, since an AS was modeled by a single node. We now take the internal router structure of ASes into account. Hence, each network node represents a router, and we assume that an IBGP session exists between every pair of routers in the same AS, leading to a network link from every router to every other router in same AS.

We have $L = \{c, r, p\} \cup (\{t\} \times \mathbb{R}^+), \Sigma = (\{\epsilon, c, r, p\} \times \mathbb{R}_0^+) \cup \{\phi\}$, and $W = (\{0, 1, 2\} \times \mathbb{R}_0^+) \cup \{+\infty\}$. The pairs of W are lexicographically ordered based on the order $\leq$. Links joining a node to another in the same AS are called internal links, and have label of the form (t, x), where x is positive and represents the IGP distance between the node at the head of the link and the node at its tail. The other links are labeled in line with the previous section: links joining a node in a provider to a node in a customer are called customer links, and have label c; links joining a node in a customer to a node in a provider are called provider links, and have label p; and links joining two nodes in different peers are called peer links, and have label r. We again distinguish four classes of paths: local paths, which are either trivial or composed of a single internal link, have signature of the form (ϵ, y); customer paths, whose first non-internal link is a customer link, have signature of the form (c, y); provider paths, whose first non-internal link is a provider link, have signature of the form (p, y); peer paths, whose first non-internal link is a peer link, have signature of the form (r, y). Moreover, paths whose first link is not internal are called external paths, their signatures having zero for second component; paths whose first link is internal are called internal paths, their signatures having a positive number for second component.

The binary operation $\oplus$ is given in Chart (1), shown at the bottom of the page, where y is a positive number. The last three rows in the chart, corresponding to labels c, r, and p, are similar to the ones in the previous section, whereas the first row corresponds to internal links. For example, $(t, x) \oplus (c, 0) = (c, x)$

$\oplus$	$(\epsilon, 0)$	(ϵ, y)	$(c, 0)$	(c, y)	$(r, 0)$	(r, y)	$(p, 0)$	(p, y)
(t, x)	(ϵ, x)	ϕ	(c, x)	ϕ	(r, x)	ϕ	(p, x)	ϕ
c	$(c, 0)$	$(c, 0)$	$(c, 0)$	$(c, 0)$	ϕ	ϕ	ϕ	ϕ
r	$(r, 0)$	$(r, 0)$	$(r, 0)$	$(r, 0)$	ϕ	ϕ	ϕ	ϕ
p	$(p, 0)$	$(p, 0)$	$(p, 0)$	$(p, 0)$	$(p, 0)$	$(p, 0)$	$(p, 0)$	$(p, 0)$

(1)

means that an external customer path becomes an internal customer path when extended by an internal link. The sequence of equalities $(t,x)\oplus(c,y)=(t,x)\oplus(r,y)=(t,x)\oplus(p,y)=\phi$, for $y>0$, means that an internal path cannot be further extended by an internal link. In other words, a router does not export to other routers in the same AS routes learned via IBGP. The usable paths in this algebra are the usable paths of the previous algebra possibly interspersed by non-consecutive internal links. The function f is given by

$$\begin{aligned} f((\epsilon,y)) &= (0,y), \quad f((c,y))=(1,y),\\ f((r,y)) &= f((p,y))=(2,y), \quad f(\phi)=+\infty. \end{aligned}$$

For instance, $f((c,0))=(1,0)\prec(1,y)=f((c,y))$ for $y>0$ means that external customer paths are preferred to internal customer paths. That is to say, routers prefer customer routes learned via EBGP to customer routes learned via IBGP.

The algebra is neither monotone nor isotone. Nevertheless, with insight acquired from the previous section, we could guess that the non-free cycles are: i) those consisting of customer links and internal links, or ii) those consisting of provider links and internal links, with the additional restriction that internal links do not appear consecutively. A formal proof of this fact is instructive in its own right, and we sketch it in the remainder of this section.

First, we show that any cycle $C=u_n u_{n-1}\cdots u_1 u_0$, with $u_n=u_0$, comprising only customer links and non-consecutive internal links is non-free. Choose the set of signatures $\alpha_0,\alpha_1,\ldots,\alpha_{n-1},\alpha_n\in\Sigma-\{\phi\}$, with $\alpha_0=\alpha_n$, as follows: if $l(u_i,u_{i-1})=c$, then $\alpha_i=(c,0)$; otherwise, if $l(u_i,u_{i-1})=(t,x_i)$, then $\alpha_i=(c,x_i)$, and because internal links do not appear consecutively, we must have $\alpha_{i-1}=(c,0)$. Therefore, if $l(u_i,u_{i-1})=c$, then $f(\alpha_i)=(1,0)=f(c\oplus\alpha_{i-1})=f(S(u_i u_{i-1},\alpha_{i-1}))$; otherwise, if $l(u_i,u_{i-1})=(t,x_i)$, then $f(\alpha_i)=(1,x_i)=f((t,x_i)\oplus(c,0))=f(S(u_i u_{i-1},\alpha_{i-1}))$. These n equalities imply that C is not free. The proof that cycles with only provider links and non-consecutive internal links are also non-free would proceed similarly.

Second, we show that all cycles other than those with only customer links and non-consecutive internal links, or those with only provider links and non-consecutive internal links, are free. Construct an auxiliary algebra which has the same elements as the original algebra except for the binary operation $\oplus'$ which is defined by the following chart.

$\oplus'$	(ϵ,y)	(c,y)	(r,y)	(p,y)
(t,x)	$(\epsilon,x+y)$	$(c,x+y)$	$(r,x+y)$	$(p,x+y)$
c	$(c,0)$	$(c,0)$	ϕ	ϕ
r	$(r,0)$	$(r,0)$	ϕ	ϕ
p	$(p,0)$	$(p,0)$	$(p,0)$	$(p,0)$

Because $f(l\oplus\alpha)\succeq f(l\oplus'\alpha)$, for all $l\in L$ and all $\alpha\in\Sigma$, every cycle that is free in the auxiliary algebra is also free in the original algebra. Since the auxiliary algebra is isotone, its free and non-free cycles can easily be determined from Theorem 4. A cycle is non-free in the auxiliary algebra if and only if (i) it has at least one customer link and all other links are either customer or internal links, or (ii) it has at least one provider link and all other links are either provider or internal links. To complete the proof, we need only show that of the non-free cycles in the auxiliary algebra those that contain at least two consecutive internal links are indeed free in the original algebra. Let $C=u_n u_{n-1}\cdots u_1 u_0$, with $u_n=u_0$, be a cycle with at least one customer link, two consecutive internal links, and with all other links either customer or internal links. Without loss of generality, assume that (u_n,u_{n-1}) is a customer link, (u_i,u_{i-1}) is an internal link with label (t,x_i), for $1\le i\le j$, with $2\le j<n$, and (u_{j+1},u_j) is again a customer link. Thus, $u_j u_{j-1}\cdots u_1 u_0$ is a subpath of cycle C with $j\ge 2$ consecutive internal links. If cycle C were not free, then there would exist $\alpha_0,\alpha_1,\ldots,\alpha_j\in\Sigma-\{\phi\}$ such that

$$\begin{aligned} &f(\alpha_0)\succeq(1,0)\\ &f(\alpha_i)\succeq f((t,x_i)\oplus\alpha_{i-1}) \quad \text{for} \quad 1\le i\le j\\ &(2,0)\succ f(\alpha_j) \end{aligned}$$

where the first and last inequalities are justified because (u_n,u_{n-1}) and (u_{j+1},u_j), respectively, are customer links. This set of $j+2\ge 4$ inequalities does not admit a solution in the original algebra. Hence, cycle C is free. Analogous methods could be used to show that cycles with at least one provider link, two consecutive internal links, and with all other links either provider or internal links are also free.

D. Backup Relationships in Interdomain Routing

We now explore backup relationships between Internet domains, resorting to the one node per AS model of Section IX-B. Backup relationships expand the set of usable paths beyond primary paths so as to maintain network connectivity in the presence of link failures. For example, if links $(6,5)$ and $(3,0)$ are down in the network of Fig. 5, then the parsimonious relationships of Section IX-B isolate node 6 from node 0. In contrast, the backup relationships of this section still allow node 6 to reach node 0 over paths 6 3 2 0, 6 4 1 0, or 6 4 3 2 0. We will call *backup paths* to usable paths that are not primary. Backup strategies have been presented in [30] and formulated in algebraic terms in our previous work [18]. Those strategies permit valleys, if they contain peer links. A *valley* is a path that starts with a customer link and ends with a provider link, implying that customer nodes may provide transit service between their providers. For example, the backup strategies of [30] would allow paths 3 0 1 4 and 4 1 0 3 in the network of Fig. 5, meaning that nodes 0 and 1 would provide transit service to their respective providers 3 and 4. In this section, we present alternative backup strategies satisfying the following requirements:

- primary paths are always preferred to backup paths;
- valleys are not allowed;
- backup paths without provider links are always preferred to those that have them;
- the preference of a backup path decreases with every peer link that it contains.

We have $L=\{c,p\}\cup(\{r\}\times\mathbb{R}^+),\Sigma=\{\epsilon,c,p,\phi\}\cup(\{r,\hat{c},\hat{p}\}\times\mathbb{R}^+)$, and $W=(\{0,1,2,3,4\}\times\mathbb{R}_0^+)\cup\{+\infty\}$. The pairs of W are lexicographically ordered based on the order $\le$. In labels, the letters c,r, and p, again identify customer, peer, and provider links, respectively. The value of x in a label of the form (r,x) is positive and corresponds to the contribution of a peer link to the avoidance level of a backup path. The *avoidance level* of a path is such that the lower its value the "better." In signatures, the letters c, p, and r identify customer, provider, and peer paths, respectively, and the accented letters $\hat{c}$ and $\hat{p}$ identify backup paths without and with provider links, respectively. The value of x in signatures of the form $(\hat{c}, x)$ and $(\hat{p}, x)$ indicates the avoidance level of a backup path. Every trivial path has signature ϵ.

The binary operation $\oplus$ operation is given next (the column for signature ϵ equals that for signature c and is omitted).

$\oplus$	c	(r,x)	p	$(\hat{c},x)$	$(\hat{p},x)$
c	c	$(\hat{c},x)$	ϕ	$(\hat{c},x)$	ϕ
(r,y)	(r,y)	$(\hat{c},x+y)$	$(\hat{p},y)$	$(\hat{c},x+y)$	$(\hat{p},x+y)$
p	p	p	p	$(\hat{p},x)$	$(\hat{p},x)$

For example, $c \oplus p = c \oplus (\hat{p}, x) = \phi$ means that a path containing provider links can never be extended by a customer link, thereby implying that valleys are not allowed. The equality $(r, y) \oplus (\hat{p}, x) = (\hat{p}, x + y)$ means that a backup path with provider links sees its avoidance level increase as it is extended by a peer link. The function f is given by

$$\begin{gathered} f(\epsilon) = (0,0), \quad f(c) = (1,0), \\ f((r,x)) = f(p) = (2,0), \quad f((\hat{c},x)) = (3,x), \\ f((\hat{p},x)) = (4,x), \quad f(\phi) = +\infty. \end{gathered}$$

For instance, any of the weights $(0,0) = f(\epsilon), (1,0) = f(c)$, and $(2,0) = f((r,x)) = f(p)$ is lexicographically smaller than both $(3,x) = f((\hat{c},x))$ and $(4,x) = f((\hat{p},x))$, meaning that primary paths are preferred to backup paths.

The algebra is monotone, it is not isotone, and its free cycles are the same as those of Section IX-B: cycles where all links have label c or all links have label p should not be present.

E. Route Reflection

Refs. [16] and [17] present examples of internal AS route reflection configurations for which IBGP does not converge. In this section, we apply the algebraic framework to arrive at a sufficient condition that ensures IBGP convergence in ASes that use route reflection. We assume a given destination outside the AS which can be reached through a number of border routers inside the AS at the same level of external preference. Each router in the AS has a unique integer identifier.

We have $L = \{d, o, u\} \times \mathbb{N}, \Sigma = (\{d, o\} \times \mathbb{N} \times \mathbb{N}) \cup (\{0, +\infty\} \times \mathbb{N}) \cup \{\phi\}$, and $W = (\mathbb{R}_0^+ \cup \{+\infty\}) \times (\mathbb{N} \cup \{+\infty\})$. The pairs of W are lexicographically ordered based on the order $\leq$. The second component in the label of each link is always the identifier of the node at the head of the link. A link that joins a route reflector to a client has d for first label component; a link that joins a route reflector to another route reflector has o for first label component; and a link with a client at its head has u for first label component. The last component in the signature of a path is always the identity of its border router. Trivial paths, those consisting of a border router alone, have signatures of the form $(0, k)$; non-trivial paths with origin at a client have signatures of the form $(+\infty, k)$; non-trivial paths with origin at a route reflector have signatures either of the form (d, i, k) or of the form (o, i, k), where i is the identity of the route reflector. Fig. 6 depicts an AS that uses route reflection, and where the border routers, for an unspecified destination outside the AS, are shaded. Path 0 is a trivial path and has signature $(0,0)$; path 6 7 2 0 has signature $(+\infty, 0)$; path 2 0 has signature $(d, 2, 0)$; path 4 2 0 has signature $(o, 4, 0)$; and path 7 4 2 0 is not usable.

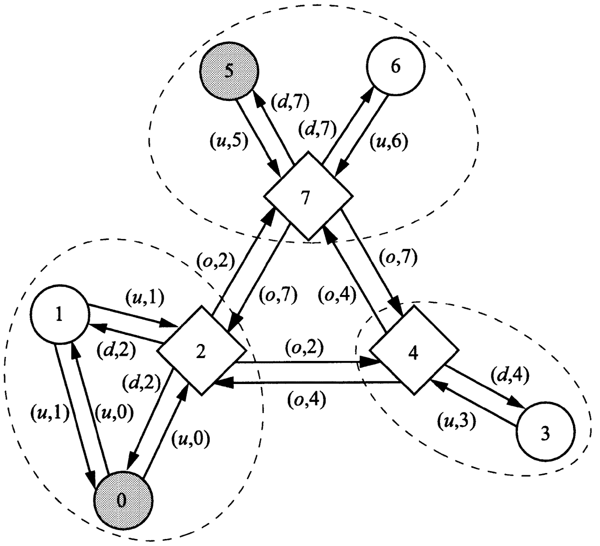

Fig. 6. AS with three clusters. Clusters are enclosed in ovals. Route reflectors are represented with diamonds, clients are represented with circles, and border routers (leading to an unspecified destination outside the AS) are shaded.

The binary operation $\oplus$ is given next.

$\oplus$	$(0,k)$	(d,i,k)	(o,i,k)	$(+\infty,k)$
(d,j)	(d,j,k)	ϕ	ϕ	ϕ
(o,j)	(o,j,k)	(o,j,k)	ϕ	ϕ
(u,j)	$(+\infty,k)$	$(+\infty,k)$	$(+\infty,k)$	ϕ

We look into some examples: $(o, j) \oplus (o, i, k) = \phi$ means that a route reflector does not export paths learned from route reflectors to other route reflectors; $(o, j) \oplus (d, i, k) = (o, j, k)$ means that route reflector i exports to route reflector j paths learned from its client border k, and the resulting path keeps the identity of the border router but sees the origin of the path updated from i to j. The function f is given next.

$$\begin{gathered} f((0,k)) = (0,k), \\ f((d,i,k)) = f((o,i,k)) = (dist(i,k),k), \\ f((+\infty,k)) = (+\infty,k), \qquad f(\phi) = (+\infty,+\infty) \end{gathered}$$

where $dist(i, k)$ is the IGP path distance from router i to router k. Forcing the algebra to be monotone leads to a sufficient condition for IBGP convergence. Monotonicity clearly holds when a trivial path is extended to any router and when any path is extended to a client. The interesting case is when a path consisting of a route reflector followed by a client border router is extended to another route reflector. The weight of the original path is $(dist(i, k), k)$, where i is the identity of the route reflector and k is the identity of its client border router. The weight of the extended path is $(dist(j, k), k)$, where j is the identity of the route reflector to which the original path has been extended. Therefore, for monotonicity to hold, we must have $dist(i, k) \leq dist(j, k)$. Moreover, if this condition is satisfied, then no free cycle is realizable in the AS because border routers have unique identifiers.

We can then conclude with generality that IBGP converges if for every client k and every route reflector j we have

$$dist(reflect(k), k) \leq dist(j, k)$$

where $reflect(k)$ is the identity of the route reflector that belongs to the same cluster as client k. In words, client k must not be farther from its route reflector $reflect(k)$ than from any other route reflector, in terms of IGP path distances.

X. Conclusion

We have brought non-classic algebraic concepts to dynamic, distributed network routing, establishing fundamental results on the convergence of routing protocols and uniting in a common framework various routing strategies currently found in the Internet. Freeness is a property of labeled networks that implies convergence of routing protocols. Monotonicity and isotonicity are two algebraic properties that strengthen the convergence properties of the protocols. Monotonicity implies protocol convergence in every network and isotonicity implies convergence onto optimal paths.

The theory has been applied to both intra- and inter-domain routing, but it is to the latter routing paradigm that it is deemed more useful in the short-term. An algebra for routing is a concise and precise mathematical object for the specification, design, and verification of routing policies.

Two standing problems in inter-domain routing deserve further study, to which the algebraic theory may be useful. One is the use of the MED attribute of BGP, which prevents the set of available routes at a router from being totally ordered by preference [31]. The other is the forwarding correctness of IBGP: guarantying IBGP convergence may not be enough to prevent packet loops [16].

Looking still further ahead, we envisage that the algebraic theory may also be expanded to incorporate traffic-aware routing and traffic engineering. By formulating non-linear routing problems with abstract algebras, we may arrive at algorithmic solutions which parallel those of linear problems [21]. In short, this paper is but a first step toward an algebraic theory of network routing.

Appendix
Proof of Convergence

Suppose that we are given a network G and a finite set $\hat{\mathcal{P}}$ of usable couplets all with the same destination. The *couplets digraph* associated with G and $\hat{\mathcal{P}}$ has the couplets of $\hat{\mathcal{P}}$ for vertices and there is an edge from couplet (Q, β) to couplet (P, α) if any one of the next two conditions is verified:

- Q is an extension of P by some link in the network, that is, $(Q, \beta) = (uv \circ P, l(u, v) \oplus \alpha)$ for some link (u, v);
- Q and P have the same origin, and either (P, α) weighs less than (Q, β) or, their weights being equal, (P, α) is preferred to (Q, β), that is, $(P, \alpha) \lhd (Q, \beta)$.

The top part of Fig. 7 shows the network of Fig. 5, annotated at each node by a list of couplets. Each list contains all couplets of the form $(P, s(P))$, where P is a usable path in the network with origin at the node by the list and destination at node 0, ordered by $\lhd$. The higher a couplet in a list the smaller it is with respect to the order $\lhd$. The bottom part of the figure shows the couplets digraph associated with the network and the couplets depicted on the top part of the figure.

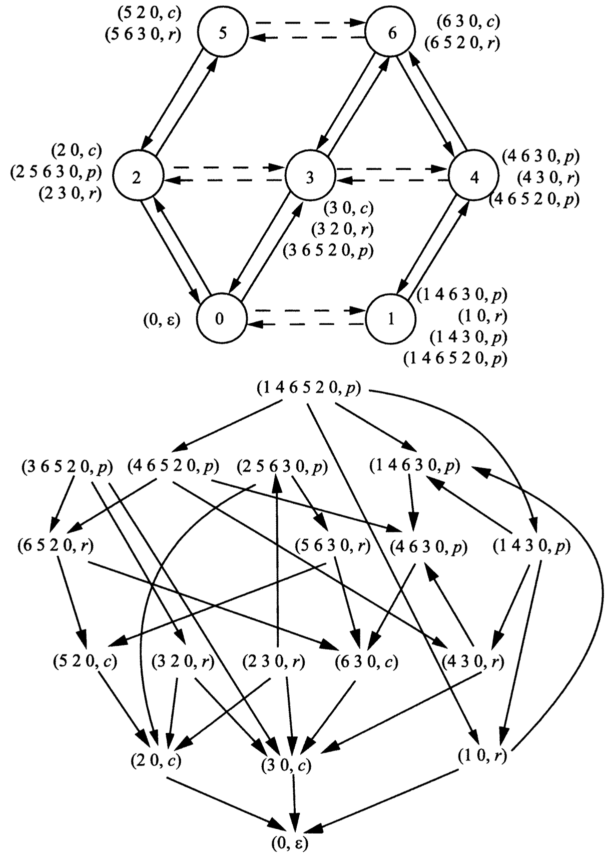

Fig. 7. Example couplets digraph for the customer-provider and peer-peer algebra of Section IX-B, network of Fig. 5, and couplets listed in the top part of the figure.

Lemma 1: Any couplets digraph associated with a free network is acyclic.

Proof: We are given a couplets digraph associated with a free network. We will prove by contradiction that the digraph is acyclic. Suppose otherwise, and let

$$\mathcal{C} = (P_m, \alpha_m)(P_{m-1}, \alpha_{m-1}) \cdots (P_1, \alpha_1)(P_0, \alpha_0)$$

with $(P_m, \alpha_m) = (P_0, \alpha_0)$, be a cycle with a minimal number of vertices. Let u_j be the origin of P_j, for $0 \leq j \leq m$. By definition, if $u_j \neq u_{j-1}$, then $P_j = u_j u_{j-1} \circ P_{j-1}$ and $f(\alpha_j) = f(l(u_j, u_{j-1}) \oplus \alpha_{j-1})$; otherwise, if $u_j = u_{j-1}$, then $(P_{j-1}, \alpha_{j-1}) \lhd (P_j, \alpha_j)$, implying $f(\alpha_j) \succeq f(\alpha_{j-1})$.

First, we show that any two couplets in cycle $\mathcal{C}$ with the same origin must appear consecutively. Suppose otherwise. Then, there are integers j and q such that $u_q = u_j, 0 \leq j, q \leq m$, and $2 \leq q - j \leq m - 2$. If $(P_q, \alpha_q) \lhd (P_j, \alpha_j)$, then the sequence

$$(P_j, \alpha_j)(P_q, \alpha_q)(P_{q-1}, \alpha_{q-1}) \cdots (P_{j+1}, \alpha_{j+1})(P_j, \alpha_j)$$

is a cycle in the couplets digraph with $q - j + 1 \leq m - 1$ vertices, contradicting the minimality of $\mathcal{C}$. On the other hand, if $(P_j, \alpha_j) \lhd (P_q, \alpha_q)$, then the sequence

$$(P_m, \alpha_m) \cdots (P_q, \alpha_q)(P_j, \alpha_j) \cdots (P_1, \alpha_1)(P_0, \alpha_0)$$

is a cycle in the couplets digraph with $m - q + j + 1 \leq m - 1$ vertices, again contradicting the minimality of $\mathcal{C}$.

TABLE I
RANKING FOR THE COUPLETS DIGRAPH OF FIG. 7

rank 1	rank 2	rank 3	rank 4	rank 5	rank 6	rank 7	rank 8
$(0,\epsilon)$	$(2\,0,c)$ $(3\,0,c)$	$(3\,2\,0,r)$ $(5\,2\,0,c)$ $(6\,3\,0,c)$	$(4\,6\,3\,0,p)$ $(5\,6\,3\,0,c)$ $(6\,5\,2\,0,r)$	$(4\,3\,0,r)$ $(1\,4\,6\,3\,0,p)$ $(2\,5\,6\,3\,0,p)$ $(3\,6\,5\,2\,0,p)$	$(1\,0,r)$ $(2\,3\,0,r)$ $(4\,6\,5\,2\,0,p)$	$(1\,4\,3\,0,p)$	$(1\,4\,6\,5\,2\,0,p)$

Second and last, we show that the sequence of nodes obtained from $u_m u_{m-1} \cdots u_1 u_0$ by skipping over repeated nodes is a network cycle which is non-free. Formally, let n be the number of distinct nodes in the sequence $u_m u_{m-1} \cdots u_1 u_0$, and define the function a from $\{0,\ldots,n\}$ to $\{0,\ldots,m\}$ as follows:

$$a(i) = \begin{cases} 0, & \text{if } i=0 \quad \text{and} \quad u_1 \neq u_0 \\ 1, & \text{if } i=0 \quad \text{and} \quad u_1 = u_0 \\ a(i-1)+1, & \text{if } 1 \le i \le n \quad \text{and} \\ & \quad u_{a(i-1)+2} \neq u_{a(i-1)+1} \\ a(i-1)+2, & \text{if } 1 \le i \le n \quad \text{and} \\ & \quad u_{a(i-1)+2} = u_{a(i-1)+1}. \end{cases}$$

The sequence $u_{a(n)} u_{a(n-1)} \cdots u_{a(1)} u_{a(0)}$ is a cycle in the network. If $a(i-1) = a(i) - 1$, with $1 \le i \le n$, then

$$\begin{aligned} f(\alpha_{a(i)}) &= f(l(u_{a(i)}, u_{a(i)-1}) \oplus \alpha_{a(i)-1}) \\ &= f(S(u_{a(i)} u_{a(i-1)}, \alpha_{a(i-1)})) \end{aligned}$$

otherwise, if $a(i-1) \neq a(i) - 1$, with $1 \le i \le n$, then $a(i-1) = a(i) - 2$, and

$$\begin{aligned} f(\alpha_{a(i)}) &\succeq f(\alpha_{a(i)-1}) \\ &= f(l(u_{a(i)}, u_{a(i)-2}) \oplus \alpha_{a(i)-2}) \\ &= f(S(u_{a(i)} u_{a(i-1)}, \alpha_{a(i-1)})). \end{aligned}$$

The previous equalities and inequalities mean that cycle $u_{a(n)} u_{a(n-1)} \cdots u_{a(1)} u_{a(0)}$ is not free, contradicting the hypothesis of the lemma. ■

Any acyclic digraph can have its vertices ranked such that if there is an edge from vertex A to vertex B, then A is ranked higher than B [19]. To be concrete, we define the *rank* of couplet (P, α) in an acyclic couplets digraph to be the number of vertices in a longest path in the digraph from (P, α) to a couplet without out-neighbors. These ranks could be determined recursively, noting that the couplets of rank j are those that have no out-neighbors in the restriction of the digraph obtained by withdrawing all couplets with rank less than j. Table I shows the ranking of the couplets of Fig. 7. In an acyclic couplets digraph, if either $(Q, \beta) = (uv \circ P, l(u,v) \oplus P)$ or $(P, \alpha) \lhd (Q, \beta)$, then (Q, β) is ranked higher than (P, α).

Theorem 6 (Liveness): The path-vector protocol always converges in a free network.

Proof: We are told that at a given time t the links interconnecting the nodes form a free network and that none of those links fail nor are new links added thereafter. We want to show that there is a subsequent instant of time when no signaling routing messages are to be found in transit in the network. The protocol will have converged at that time.

We fix an arbitrary destination d and prove convergence of the protocol for that destination. Observing the state of the protocol at time t, we find usable couplets stored at the pairs of variables $(ptab_u[v,d], stab_u[v,d])$, with u a network node and v an out-neighbor of u, and announced in signaling routing messages, which are in transit in the network. Let $\hat{\mathcal{P}}_{\text{ini}}$ be the set of all such couplets. We assume that $\hat{\mathcal{P}}_{\text{ini}}$ is finite. This is a mild assumption that holds if, for instance, the number of events—link failures, link additions, and receptions of signaling routing messages—is finite up to time t. Any usable couplet that can possibly be found in the state of the protocol after time t has to be of the form $(P' \circ P, S(P', \alpha))$, where P' is a path in the network and (P, α) is in $\hat{\mathcal{P}}_{\text{ini}}$. Let $\hat{\mathcal{P}}$ be the set of all such couplets. Set $\hat{\mathcal{P}}$ is finite, because $\hat{\mathcal{P}}_{\text{ini}}$ is finite and so is the set of all paths in the network. Since the network is acyclic, the couplets of $\hat{\mathcal{P}}$ are ranked. Let M be the maximum rank assigned to a couplet of $\hat{\mathcal{P}}$. By definition, couplet $(none, \phi)$ is assigned rank $M+1$.

We now present a function F from the state of the protocol to the well-founded set of $M+1$ tuples of non-negative integers ordered lexicographically. We show that the value assumed by F decreases lexicographically with the reception of every signaling routing message, and this is sufficient to prove convergence of the protocol [24], [25]. At any given time, the value assumed by the jth coordinate of the function F is denoted by f_j and is defined as:

$f_j =$ number of signaling routing messages announcing a couplet of rank j, in transit in the network, **plus** number of nodes that have chosen a couplet of rank j.

Assume that a signaling routing message announcing couplet (P, α), of rank j, arrives at node u coming from its out-neighbor v. Let (Q, β), a couplet of rank k, be the chosen couplet at node u before the routing message is received, and let (R, γ), a couplet of rank l, be the chosen couplet at node u after the routing message is received. Four cases are distinguished.

1) $(R, \gamma) = (Q, \beta)$: The coordinate f_j decreases by one. The function F decreases.
2) $(R, \gamma) \neq (Q, \beta)$ and $(R, \gamma) = (uv \circ P, l(u,v) \oplus \alpha)$: The coordinate f_j decreases by one, the coordinate f_k decreases by one, and the coordinate f_l increases. Because $(R, \gamma) = (uv \circ P, l(u,v) \oplus \alpha)$, we have $l > j$ and, therefore, the function F decreases.
3) $(R, \gamma) \neq (Q, \beta)$ and $(R, \gamma) = (none, \phi)$: The coordinate f_j may decrease by one, the coordinate f_k decreases by one, and the coordinate f_{M+1} may increase. Because $(R, \gamma) \neq (Q, \beta)$ and $R = (none, \phi)$, we have $k < M+1$, and the function F decreases.
4) $(R, \gamma) \neq (Q, \beta)$ and $(R, \gamma) \neq (none, \phi)$ and $(R, \gamma) \neq (uv \circ P, l(u,v) \oplus \alpha)$: The coordinate f_j decreases by one, the coordinate f_k decreases by one, and the coordinate f_l increases. Because $(R, \gamma) \neq (uv \circ P, l(u,v) \oplus \alpha)$, couplet (R, γ) was available for selection at node u before the signaling routing message was received. Since, in addition, $(R, \gamma) \neq (Q, \beta)$ and couplet (Q, β) was the couplet chosen by u before the signaling routing message is

received, we have $(Q, \beta) \lhd (R, \gamma)$. Hence, $k < l$ and the function F decreases. ■

Theorem 7 (Safety): If the network is free and the path-vector protocol has converged, it has converged onto local-optimal in-trees.

Proof: Once the protocol has converged there are no more signaling routing messages in transit in the network. Because signaling routing messages are never lost and are delivered according to a first-in-first-out discipline, we have, for every link (u, v) in the network:

- if u is not a node of $path_v[d]$ and $l(u, v) \oplus sign_v[d] \neq \phi$, then $(ptab_u[v, d], stab_u[v, d]) = (uv \circ path_v[d], l(u, v) \oplus sign_v[d])$;
- otherwise, if either u is a node of $path_v[d]$ or $l(u, v) \oplus sign_v[d] = \phi$, then $(ptab_u[v, d], stab_u[v, d]) = (none, \phi)$.

Fix an arbitrary destination d. Using freeness of the network, it can readily be shown that the union of the paths $path_u[d]$ over all network nodes u such that $path_u[d] \neq none$ is an in-tree rooted at destination d, with the signatures of the in-tree paths given by $s(path_u[d]) = sign_u[d]$. We need to show that this in-tree is local-optimal.

Consider first a node u outside the in-tree, and let v be any out-neighbor of u in the in-tree. Path $path_v[d]$ does not contain u because, otherwise, $path_u[d] \neq none$, meaning that node u would belong to the in-tree. Then, we must have $s(uv \circ path_v[d]) = l(u, v) \oplus s(path_v[d]) = \phi$, meaning that $uv \circ path_v[d]$ is not a usable path from u to d.

Second, consider nodes u and v in the in-tree, with v any out-neighbor of u. If path $path_v[d]$ does not contain u, then directly from the path-vector protocol code, we know that

$$f(s(path_u[d])) \preceq f(s(uv \circ path_v[d])).$$

On the other hand, if path $path_v[d]$ contains u, then we can write $path_v[d] = u_n u_{n-1} \cdots u_1 u_0 \circ path_u[d]$, with $u_0 = u$ and $u_n = v$. The sequence $u_0 u_n \cdots u_1 u_0$ is a cycle in the network which must, by our hypothesis, be free. Define $\alpha_i = s(u_i u_{i-1} \cdots u_1 u_0 \circ path_u[d])$ for $0 \leq i \leq n$. Hence, $f(\alpha_i) = f(S(u_i u_{i-1}, \alpha_{i-1}))$ for $1 \leq i \leq n$. Freeness then implies

$$\begin{aligned} f(s(path_u[d])) &= f(\alpha_0) \\ &\prec f(S(uv, \alpha_n)) \\ &= f(s(uv \circ path_v[d])). \end{aligned}$$

Therefore, path $path_u[d]$ weighs less than walk $uv \circ path_v[d]$, thereby concluding the proof. ■

References

[1] G. Malkin, “RIP Version 2,” RFC 2453, Nov. 1998.

[2] G. Malkin, *RIP: An Intra-Domain Routing Protocol*. Reading, MA: Addison Wesley, 1999.

[3] J. Doyle, *Routing TCP/IP*. Indianapolis, IN: Cisco Press, 1998.

[4] J. T. Moy, “OSPF Version 2,” RFC 2328, Apr. 1998.

[5] J. T. Moy, *OSPF: Anatomy of an Internet Routing Protocol*. Reading, MA: Addison Wesley, 1998.

[6] C. Huitema, *Routing in the Internet*, 2nd ed. Englewood Cliffs, NJ: Prentice Hall PTR, 2000.

[7] Y. Rekhter and T. Li, “A Border Gateway Protocol 4 (BGP-4),” RFC 1771, Mar. 1995.

[8] J. W. Stewart III, *BGP4: Inter-Domain Routing in the Internet*. Reading, MA: Addison Wesley, 1999.

[9] J. L. Sobrinho, “Algebra and algorithms for QoS path computation and hop-by-hop routing in the Internet,” *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, pp. 541–550, Aug. 2002.

[10] D. P. Bertsekas and R. Gallager, *Data Networks*, 2nd ed: Prentice-Hall, 1991.

[11] M. G. Gouda and M. Schneider, “Maximizable routing metrics,” *IEEE/ACM Trans. Netw.*, vol. 11, no. 4, pp. 663–675, Aug. 2003.

[12] L. Gao and J. Rexford, “Stable Internet routing without global coordination,” *IEEE/ACM Trans. Netw.*, vol. 9, no. 6, pp. 681–692, Dec. 2001.

[13] T. Bates and R. Chandra, “BGP Route Reflection: An Alternative to Full Mesh IBGP,” RFC 1966, Jun. 1996.

[14] R. Dube, “A comparison of scaling techniques for BGP,” *Comput. Commun. Rev.*, vol. 29, no. 3, pp. 44–46, 1999.

[15] J. G. Scudder and R. Dube, “BGP scaling techniques revisited,” *Comput. Commun. Rev.*, vol. 29, no. 5, pp. 22–23, Oct. 1999.

[16] T. Griffin and G. Wilfong, “On the correctness of IBGP configuration,” in *Proc. ACM SIGCOMM*, Pittsburgh, PA, Aug. 2002, pp. 17–29.

[17] A. Basu, C.-H. Ong, A. Rasala, F. Shepherd, and G. Wilfong, “Route oscillations in I-BGP with route reflection,” in *Proc. ACM SIGCOMM*, Pittsburgh, PA, Aug. 2002, pp. 235–247.

[18] J. L. Sobrinho, “Network routing with path vector protocols: Theory and applications,” in *Proc. ACM SIGCOMM*, Karlsruhe, Germany, Aug. 2003, pp. 49–60.

[19] B. Carré, *Graphs and Networks*. Oxford, U.K.: Clarendon Press, 1979.

[20] M. Gondran and M. Minoux, *Graphes et Algorithmes*, 3rd ed. Paris, France: Eyrolles, 1995.

[21] M. Gondran and M. Minoux, *Graphes, Dioïdes et Semi-Anneaux*. Paris, France: Editions Tec & Doc, 2001.

[22] T. Griffin, F. Shepherd, and G. Wilfong, “Policy disputes in path-vector protocols,” in *Proc. 7th Int. Conf. Network Protocols*, Toronto, ON, Canada, Nov. 1999, pp. 21–30.

[23] T. Griffin, F. Shepherd, and G. Wilfong, “The stable paths problem and interdomain routing,” *IEEE/ACM Trans. Netw.*, vol. 10, no. 2, pp. 232–243, Apr. 2002.

[24] L. Lamport, “An assertional correctness proof of a distributed algorithm,” *Sci. Comput. Program.*, vol. 2, no. 3, pp. 175–206, Dec. 1982.

[25] L. Lamport, “The temporal logic of actions,” *ACM Trans. Program. Lang. Syst.*, vol. 16, no. 3, pp. 872–923, Apr. 1994.

[26] T. Griffin, A. Jaggard, and V. Ramachandran, “Design principles of policy languages for path vector protocols,” in *Proc. ACM SIGCOMM*, Karlsruhe, Germany, Aug. 2003, pp. 61–72.

[27] C. Alaettinoğlu, “Scalable router configuration for the Internet,” in *Proc. 5th Int. Conf. Computer Communications and Networks*, Rockville, MD, Oct. 1996, pp. 325–328.

[28] G. Huston, “Interconnections, peering and financial settlements,” in *Proc. INET’99*, San Jose, CA, Jun. 1999.

[29] L. Gao, “On inferring autonomous system relationships in the Internet,” *IEEE/ACM Trans. Netw.*, vol. 9, no. 6, pp. 733–745, Dec. 2001.

[30] L. Gao, T. Griffin, and J. Rexford, “Inherently safe backup routing with BGP,” in *Proc. IEEE INFOCOM*, Anchorage, AK, Apr. 2001, pp. 547–556.

[31] T. Griffin and G. Wilfong, “An analysis of the MED oscillation problem in BGP,” in *Proc. IEEE Int. Conf. Network Protocols*, Paris, France, Nov. 2002, pp. 90–99.

João Luís Sobrinho (M’97) received the Licenciatura and Ph.D. degrees in electrical and computer engineering from the Technical University of Lisbon, Portugal.

He is currently an Assistant Professor at the same University. Previously, he worked for Bell Laboratories, Lucent Technologies, in The Netherlands. His current research focuses on distributed algorithms for communication networks, with a special interest in routing.

Dr. Sobrinho has been a member of the Association for Computing Machinery (ACM) since 2003.