Network Routing with Path Vector Protocols: Theory and Applications

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

ABSTRACT

Path vector protocols are currently in the limelight, mainly because the inter-domain routing protocol of the Internet, BGP (Border Gateway Protocol), belongs to this class. In this paper, we cast the operation of path vector protocols into a broad algebraic framework and relate the convergence of the protocol, and the characteristics of the paths to which it converges, with the monotonicity and isotonicity properties of its path compositional operation. Here, monotonicity means that the weight of a path cannot decrease when it is extended, and isotonicity means that the relationship between the weights of any two paths with the same origin is preserved when both are extended to the same node. We show that path vector protocols can be made to converge for every network if and only if the algebra is monotone, and that the resulting paths selected by the nodes are optimal if and only if the algebra is isotone as well.

Many practical conclusions can be drawn from instances of the generic algebra. For performance-oriented routing, typical in intra-domain routing, we conclude that path vector protocols can be made to converge to widest or widest-shortest paths, but that the composite metric of IGRP (Interior Gateway Protocol), for example, does not guarantee convergence to optimal paths. For policy-based routing, typical in inter-domain routing, we formulate existing guidelines as instances of the generic algebra and we propose new ones. We also show how a particular instance of the algebra yields a sufficient condition for signaling correctness of internal BGP.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols—*routing protocols*

General Terms

Algorithms, Theory

SIGCOMM'03, August 25–29, 2003, Karlsruhe, Germany

Keywords

Path Vector Protocols, Algebra, Border Gateway Protocol, BGP

1. INTRODUCTION

Path vector protocols have recently attracted much attention, mostly because the only protocol for inter-domain routing in the Internet, BGP (Border Gateway Protocol) [13, 16, 9], belongs to this class of protocols. Other protocols seem to follow suit, such as those for optical inter-networking [1] and telephony routing over IP [14].

We feel that current analysis of path vector protocols have been too tied to the specifics of particular systems hindering a broad understanding of what can and cannot be accomplished with those protocols in terms of convergence and characteristics of the paths the protocols converge to. In this work, we provide a modern algebraic theory of path vector protocols. The algebra comprises a set of labels, a set of signatures, and a set of weights. There is an operation to obtain the signature of a path from the labels of its constituent links, and a function mapping signatures into weights. Ultimately, each path will have a weight, and these weights are ordered so that any set of paths with the same origin and destination can be compared. The concept of optimal path follows naturally from this framework, and we adjoin it with the more general concept of local-optimal path.

The challenge in this approach is to find exactly the primitive properties that should be imposed on the algebra so that definite and general statements about protocol convergence can be made. Monotonicity and isotonicity are the two such properties. Monotonicity means that the weight of a path does not decrease when it is extended, and isotonicity means that the relationship between the weights of any two paths with the same origin is preserved when both are extended to the same node. We conclude that path vector protocols can be made to converge robustly, for every network, if and only if the algebra is monotone. In this case, the set of paths the protocol converges to are local-optimal paths. The local-optimal paths become optimal paths if and only if the algebra is isotone as well as monotone.

Many applications can be drawn from the general theory. For environments where routing performance is the main concern, we conclude, for example, that path vector protocols can be used to make packets travel over widest or widest-shortest paths, but that the composite metric used by IGRP (Interior Gateway Routing Protocol) [2] does not make them travel over optimal paths. The most immediate

practical application of the generic framework, however, is to policy-based routing and BGP. We formulate the guidelines of Gao and Rexford [4] and Gao, Griffin and Rexford [3] in algebraic terms, showing that the first can be regarded as an optimal path problem but the latter cannot. The framework is also used to present new guidelines for policy-based routing with BGP, to discuss QoS (Quality-of-Service) extensions to BGP [17], and to derive a sufficient condition for signaling correctness of iBGP (internal BGP).

We discuss related work in the next section. The network model and some definitions are given in Section 3. The properties of the algebra and the concepts of optimal and local-optimal paths are presented in Section 4. The path vector protocol used as reference appears in Section 5, and the convergence results are stated and discussed in Section 6. Section 7 is dedicated to applications and counter-examples, leaving the proof of the main convergence result to Section 8. Section 9 discusses the use of the algebraic framework in a BGP context, just before the paper ends, in Section 10.

2. RELATED WORK

Besides the work on guidelines for policy-based routing with BGP and QoS extensions to BGP, already referred to in the introduction, our work relates with two other research areas: algebras for network routing; convergence of path vector protocols.

The application of modern algebraic concepts to network routing problems seems to have been initiated by Sobrinho [15], with a study on optimal path routing supported on link-state protocols. The algebra in the present work contemplates both optimal and local-optimal path routing and is the one algebra suited to path vector protocols, as opposed to link-state protocols.

The convergence of generic path vector protocols was first studied by Griffin, Shepherd, and Wilfong [6, 7] using a combinatorial model. In this model, the problem is represented by sets of ordered paths, one set per node, leading to a representation whose size may be exponential in the size of the network. This cardinality is carried through to the size of the data structures used to verify convergence, exacting a computational toll on such a verification. The algebraic model presented here is positioned at a higher level of abstraction than the combinatorial model bringing two main advantages. On the one hand, an algebra provides a semantic context for the design and specification of routing strategies. On the other hand, the monotonicity and isotonicity of an algebra, properties which can typically be checked at low computational complexity (see Section 6.3), completely determine the convergence properties of path vector protocols.

3. NETWORK MODEL AND TERMINOLOGY

A network is modelled as a directed graph. Given link (u, v) in the network, we say that node u is the *head* of the link, that node v is an *out-neighbor* of node u, and that node u is an *in-neighbor* of node v. In general, the presence of link (u, v) in the 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.

A path is a directed graph with node and link sets of the form $\{u_n, u_{n-1}, \cdots, u_1\}$ and $\{(u_n, u_{n-1}), \cdots, (u_2, u_1)\}$, respectively. This path is represented by $u_n u_{n-1} \cdots u_1$, with u_n and u_1 being its *origin* and *destination*, respectively. Given two paths Q and P, if their nodes are distinct except for the destination of Q and the origin of P, then their union is also a path which we denote by $Q \circ P$. In particular, if uv is a path with only two nodes, then the path $uv \circ P$ is called the *extension* of path P to node u. If link (u_1, u_n) is added to path $u_n \cdots u_2 u_1$, we obtain a *cycle*. This cycle is represented as $u_n u_{n-1} \cdots u_1 u_0$ with the understanding that $u_0 = u_n$. An *in-tree* is a directed graph with the following three properties: there is only one node, called the *root*, without out-neighbors; all nodes other than the root have one and only one out-neighbor; there is a path from every node to the root. Figure 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.

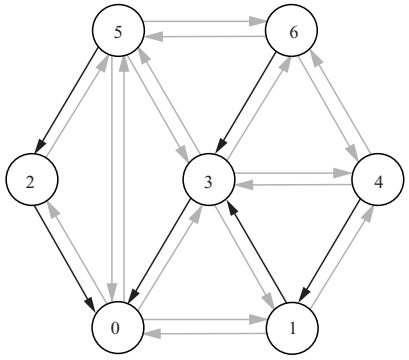

Figure 1: The dark links represent an in-tree rooted at node 0.

4. ALGEBRA

4.1 Properties

The algebra is a seven-tuple $(W, \preceq, L, \Sigma, \phi, \oplus, f)$. It comprises a set *weights*, W, a set of *labels*, L, and a set of *signatures*, Σ, with special signature ϕ. The set of weights is totally ordered by the relation $\preceq$. The operation $\oplus$ has domain $L \times \Sigma$ and range Σ, and the function f maps signatures to weights. Properties of the algebra are given next:

Maximality $\forall_{\alpha \in \Sigma - \{\phi\}} \quad f(\alpha) \prec f(\phi)$

Absorption $\forall_{l \in L} \quad l \oplus \phi = \phi$

Monotonicity $\forall_{l \in L} \forall_{\alpha \in \Sigma} \quad f(\alpha) \preceq f(l \oplus \alpha)$

Isotonicity $\forall_{l \in L} \forall_{\alpha, \beta \in \Sigma} \quad f(\alpha) \preceq f(\beta) \Rightarrow f(l \oplus \alpha) \preceq f(l \oplus \beta)$

Maximality and absorption are trivial properties, which we assume always hold. The interesting properties, on which we center our study, are monotonicity and isotonicity. Monotonicity is important for convergence of path vector protocols, and the conjunction of monotonicity and isotonicity is important for convergence to optimal paths.

The relation $\prec$ is defined such that $a \prec b$ if $a \preceq b$ and $a \neq b$, and the relation $\succ$ is defined such that $a \succ b$ if $b \prec a$. We make the distinction between monotonicity, as just defined, and its stronger kin, called strict monotonicity:

Strict monotonicity $\forall_{l \in L} \forall_{\alpha \in \Sigma - \{\phi\}} \quad f(\alpha) \prec f(l \oplus \alpha)$

4.2 Optimal and local-optimal paths

Each network link carries a label, and each network path has a signature. The label of link (u, v) is denoted by $l(u, v)$. The signature of the trivial path composed of node d alone is denoted by $s(d)$. The signature of the non-trivial path $uv \circ Q$ is defined inductively as follows:

$$s(uv \circ Q) = l(u, v) \oplus s(Q).$$

The operation s is well-defined since path Q has one less node than path $uv \circ Q$. We refer to $f(s(P))$ as the weight of path P. Monotonicity implies that $f(s(P)) \preceq f(s(Q \circ P))$, that is, the weight of a path cannot decrease when it is prefixed by another path. On the other hand, isotonicity yields that $f(s(P)) \preceq f(s(R))$ implies $f(s(Q \circ P)) \preceq f(s(Q \circ R))$, that is, the weight relationship between two paths with the same origin is preserved when both are prefixed by a common, third, path.

A path is *usable* if its signature is different from ϕ. An *optimal path* from node u to d is a usable path with weight less than or equal, according to the order $\preceq$, to the weight of any other path from u to d. An *optimal-paths in-tree* rooted at node d is an in-tree rooted at d which satisfies the next two conditions:

- if node u belongs to the in-tree, then the only path in the in-tree from u to d is an optimal path;
- if node u does not belong to the in-tree, then there is no optimal path from u to d.

Contrary to the concept of optimal path, the concept of local-optimal path from node u to node d exists only with respect to a set of paths, each with origin in an out-neighbor of node u and destination at node d. Let $\mathcal{V}$ be a set of such paths, and let $\overline{\mathcal{V}}$ be the extensions of the paths in $\mathcal{V}$ to node u.

$$\overline{\mathcal{V}} = \{uv \circ P : P \in \mathcal{V},\ u \text{ is not a node of } P\}.$$

A *local-optimal* path with respect to the set $\mathcal{V}$ is a usable path of $\overline{\mathcal{V}}$ with weight less than or equal to the weight of any other path in $\overline{\mathcal{V}}$. Given an in-tree rooted at node d, T_d, we define $\mathcal{V}_u(T_d)$ as the set of in-tree paths which have an out-neighbor of node u for origin and node d for destination. For instance, in the in-tree T_0 of Figure 1, we have $\mathcal{V}_1(T_0) = \{0, 3\ 0, 4\ 1\ 3\ 0\}$. The in-tree T_d is a *local-optimal-paths in-tree* if it satisfies the next two conditions:

- if node u belongs to the in-tree, then the only path in the in-tree from u to d is a local-optimal path with respect to $\mathcal{V}_u(T_d)$;
- if node u does not belong to the in-tree, then there is no local-optimal path from u to d with respect to $\mathcal{V}_u(T_d)$.

So, in a local-optimal-paths in-tree, the in-tree path from node u to the destination is local-optimal with respect to the in-tree paths with origin at the out-neighbors of node u.

We will now establish that, given a monotonic algebra, a local-optimal-paths in-tree is an optimal-paths in-tree if and if the algebra is isotone. First, we need the following proposition.

PROPOSITION 1. *If the algebra is isotone as well as monotone, then there is an optimal path from node u to node d such that all of its subpaths with destination at d are optimal paths on their own.*

PROOF. We sketch a proof by contradiction. Suppose the algebra is both monotone and isotone and that for every path from u to d there is a node along this path such that the subpath with origin at that node and destination at d is not an optimal path. Let $u_1u_2 \cdots u_k \circ P$ be an optimal path from $u = u_1$ to d for which $u_1u_2 \cdots u_k$ is a maximal subpath (i.e., k is maximal) such that the subpath with origin at u_i, $1 \leq i < k$, and destination at d is an optimal path, but the path P, with origin at u_k and destination at d, is not. Clearly, $u_k \neq d$. Let $u_ku_{k+1} \circ Q$ be an optimal path from u_k to d. From monotonicity, node u_i, $1 \leq i < k$, cannot be a node of Q. Hence, we can form the path $u_1u_2 \cdots u_ku_{k+1} \circ Q$. From isotonicity and $f(s(u_ku_{k+1} \circ Q)) \prec f(s(P))$, we conclude that $f(s(u_i \cdots u_ku_{k+1} \circ Q)) \preceq f(s(u_i \cdots u_k \circ P))$ for $1 \leq i < k$: the subpath $u_i \cdots u_ku_{k+1} \circ Q$ is optimal. By hypothesis so is subpath $u_ku_{k+1} \circ Q$, and this contradicts the choice of path $u_1u_2 \cdots u_k \circ P$. □

PROPOSITION 2. *Given a monotonic algebra, every local-optimal-paths in-tree is an optimal-paths in-tree if and only if the algebra is isotone.*

PROOF. We first show the direct implication. Suppose that T_d is a local-optimal-paths in-tree rooted at d which is not an optimal-paths in-tree. Then, there is a node u with an optimal path to d such that either u does not belong to T_d or the unique path in T_d from u to d is not an optimal path. Let $u_nu_{n-1} \cdots u_1$, with $u_n = u$ and $u_1 = d$, be an optimal path from u to d such that all of its subpaths with destination d are also optimal paths. The existence of this path is assured by Proposition 1. For every k such that u_k belongs to in-tree T_d, let P_k be the path in the in-tree from u_k to d. Let i, $i > 1$, be the smallest index such that either u_i does not belong to T_d or P_i is not an optimal path. In the latter case, we have $f(s(u_iu_{i-1} \cdots u_1)) \prec f(s(P_i))$. Both P_{i-1} and $u_{i-1} \cdots u_1$ are optimal paths, $P_{i-1} \in \mathcal{V}_{u_i}(T_d)$ and, from monotonicity, P_{i-1} does not contain u_i. From isotonicity, $f(s(u_{i-1} \cdots u_1)) = f(s(P_{i-1}))$ implies $f(s(u_iu_{i-1} \cdots u_1)) = f(s(u_iu_{i-1} \circ P_{i-1}))$. In particular, $s(u_iu_{i-1} \circ P_{i-1}) \neq \phi$, so that node u_i has to belong T_d. But then $f(s(u_iu_{i-1} \circ P_{i-1})) \prec f(s(P_i))$ which contradicts the assumption of P_i being a local-optimal-path with respect to $\mathcal{V}_{u_i}(T_d)$. In conclusion, T_d is an optimal-paths in-tree.

We show the converse statement with the help of Figure 2. 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 Figure 2, path P has signature α, path Q has signature β, and link (u, v) has label l. The in-tree that contains path P to the disadvantage of path Q is a local-optimal-paths in-tree because $f(s(P)) = f(\alpha) \preceq f(\beta) = f(s(Q))$. As a consequence, the path in the local-optimal-paths in-tree from u to d is path $uv \circ P$, if any. However, that is not an optimal path from u to d, since $f(s(uv \circ P)) = f(l \oplus \alpha) \succ f(l \oplus \beta) = f(s(uv \circ Q))$. □

In Section 7, we will present examples of algebras which are monotone but not isotone, for which local-optimal-paths in-trees are not necessarily optimal-paths in-trees.

5. PATH VECTOR PROTOCOL

Given a destination, each node participating in a path vector protocol chooses, at any given time, a local-optimal path with respect to the paths last learned from each of its out-neighbors to reach the destination. If there is more than

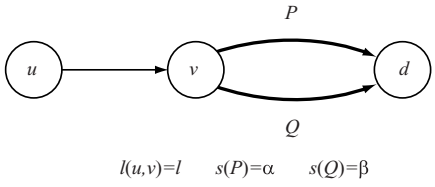

Figure 2: Thick lines represent paths and thin lines represent links. Suppose that $f(\alpha) \preceq f(\beta)$, but $f(l \oplus \alpha) \succ f(l \oplus \beta)$. Then, the local-optimal-paths in-tree rooted at d that contains path P, and not path Q, is not an optimal-paths in-tree rooted at d.

one local-optimal path, the node deterministically chooses one of them. We assume that the relative preference given to paths with the same origin, destination, and weight totally orders those paths.

Algorithm 1 presents representative path vector protocol code for node u to reach destination d. This code is executed atomically when node u receives a signaling routing message from its out-neighbor v'. A signaling routing message is of the form $\langle P, s \rangle$: if P is a path, then s is its signature; otherwise, if $P = none$, then s is equal to ϕ. The symbol $none$ denotes the absence of a path. The variable $path_u$ holds the path currently chosen at node u to reach node d, and the variable $ptab_u[v]$ holds the chosen path with origin at v and destination at d last learned from out-neighbor v. The variables $sign_u$ and $sign_u[v]$ hold the signatures of paths $path_u$ and $ptab_u[v]$, respectively. Algorithm 1 states simply that when node u receives a signaling routing message from its out-neighbor v', it updates its chosen path to the destination to become the most preferred of the local-optimal paths with respect to the paths $ptab_u[v]$, and it advertises the new chosen path to all in-neighbors, if the chosen path has changed as a result of the update. Similar code exists to deal with the failure, addition, or change of label of a link. We assume that for each pair of nodes u and v such that v is a out-neighbor of u there is a signaling queue to hold the signaling routing messages in transit from v to u. This signaling queue is lossless and behaves according to a first-in-first-out service discipline.

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 $path_u$, or that $vu \circ path_u$ is not a usable path, it may send routing message $\langle none, \phi \rangle$ to in-neighbor v, instead of routing message $\langle path_u, sign_u \rangle$. Also, the signature of a path may be omitted from the signaling routing messages 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 signaling routing message. These variations do not alter our main conclusions.

Algorithm 1 Protocol code when node u receives signaling routing message $\langle P, s \rangle$ from out-neighbor v'.

```
ptab_u[v'] := P
sign_u[v'] := s
if there is a local-optimal path with respect to the paths
ptab_u[v] then
   let uv* ∘ ptab_u[v*] be the preferred local-optimal path
   with respect to the paths ptab_u[v]
   path_u := uv* ∘ ptab_u[v*]
   sign_u := l(u, v*) ⊕ ptab_u[v*]
else
   path_u := none
   sign_u := φ
if path_u has changed then
   for all v in-neighbor of u do
      send ⟨path_u, sign_u⟩ to v
```

6. PROTOCOL CONVERGENCE

6.1 Specification

The specification of every path vector protocol contains at the very least the convergence requirement. This requirement imposes that some time after links stop failing and being added between nodes no more signaling routing messages are to be found in transit in signaling queues. Further requirements in the specification of a path vector protocol care to the properties of the paths chosen by the nodes once the protocol has converged, and these requirements depend on the particular routing strategies one wishes to implement. A generic requirement usually found in performance-oriented routing strategies is the optimality requirement, which states that the union of all paths chosen by the nodes to reach any given destination should form an optimal-paths in-tree rooted at that destination.

6.2 Main convergence results

It is easy to show that if the protocol converges, then, once it has converged, the path choices at the nodes yield local-optimal-paths in-trees rooted at the various destinations. We omit the proof because it does not depend on the monotonicity and isotonicity properties of the algebra, and because it can be adapted from a similar proof in [7]. From Proposition 2, we already know that if the algebra is monotone, then local-optimal-path in-trees are optimal-path in-trees if and only if the algebra is isotone as well. It is the relationship between convergence and monotonicity that remains to be established.

The necessity of monotonicity for protocol convergence can be shown with an example. If the algebra is not monotone, then there are $l \in L$ and $\alpha \in \Sigma$ such that $f(l \oplus \alpha) \prec f(\alpha)$. From the absorptive property, we conclude that $\alpha \neq \phi$. In the network of Figure 3, node d is the destination. Suppose that signaling routing messages incur a delay of exactly one unit of time travelling either from u to v or from v to u. At time zero, nodes u and v have just chosen paths P_u and P_v to reach node d, respectively, and advertised these choices to each other. After one unit of time as elapsed, node u learns of path P_v and, because $f(s(uv \circ P_v)) = f(l \oplus \alpha) \prec f(\alpha) = f(s(P_u))$, it changes it chosen path to $uv \circ P_v$; ditto for node v which changes its chosen path to reach d to $vu \circ P_u$. After one more unit of time has elapsed, node u learns that node v has chosen path $vu \circ P_u$ to reach d. Since this path contains node u it is not an option for node u: node u reverts its path choice to P_u. Similarly, node v reverts its path choice to P_v. We are back at the initial conditions, the described sequence of events repeats itself, and the protocol never converges. Note that, in this particular example, there are two local-optimal-paths in-trees rooted at d, despite non-convergence of the path

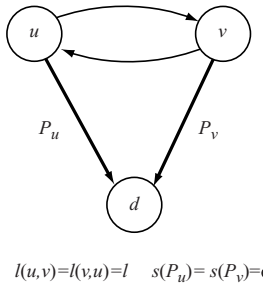

Figure 3: Thick lines represent paths and thin lines represent links. Suppose that $f(\alpha) \succ f(l \oplus \alpha)$. Then, paths $uv \circ P_v$ and $vu \circ P_u$ weigh less than paths P_u and P_v, respectively. If signaling routing messages are exchanged synchronously, then the path vector protocol never converges. The same conclusion holds if $f(\alpha) = f(l \oplus \alpha)$, but nodes u and v prefer paths $uv \circ P_v$ and $vu \circ P_u$ to paths P_u and P_v, respectively, to reach node d.

vector protocol to either of them. One is the in-tree that contains link (u, v) and path P_v, and the other is the in-tree that contains link (v, u) and path P_u.

Even if the algebra is monotone, protocol convergence depends on the relative path preferences assigned by the nodes to paths with the same weight. Suppose that the algebra is not strict monotone. Then there are $l \in L$ and $\alpha \in \Sigma - \{\phi\}$ such that $f(l \oplus \alpha) = f(\alpha)$. Let us go back to Figure 3, now with the understanding that $f(l \oplus \alpha) = f(\alpha)$. Since paths $uv \circ P_v$ and P_u have the same weight, we may assume that node u preferes the former path to the latter to reach destination d. Likewise, we may assume that node v prefers path $vu \circ P_u$ to path P_v to reach node d. With these preferences, the path choices at nodes u and v oscillate as before and the protocol never converges.

The relative preferences given to paths with the same weight become irrelevant, as far as convergence is concerned, in networks which we call *free*:

Freeness $\forall_{\text{cycle } u_n \cdots u_1 u_0} \forall_{w \in W - \{f(\phi)\}} \exists_{0<i\leq n} \forall_{\alpha \in \Sigma}$
$f(\alpha) = w \Rightarrow f(l(u_i, u_{i-1}) \oplus \alpha) \neq w.$

Taken together with monotonicity, freeness implies that given a cycle and a set of paths with origins at the nodes of the cycle, all with the same weight, at least one of these paths will see its weight increase as it extends into the cycle. Clearly, if the algebra is strict monotone, then every network is free.

PROPOSITION 3. *If the algebra is monotone and the network is free, then, whatever the relative preference given to paths with the same weight, the path vector protocol converges.*

Proposition 3 is proven in Section 8.

The question we address now is whether we can raise the condition of the network being free accepting, on the other hand, constraints on the relative path preferences given to paths with the same weight. In this regard, we have the following proposition.

PROPOSITION 4. *If the algebra is monotone and nodes prefer paths with minimum number of links among those with the same weight, then, whatever the network, the path vector protocol converges.*

PROOF. We only sketch the proof. From the algebra $(W, \preceq, L, \Sigma, \phi, \oplus, f)$, we can construct another where the number of links in a path becomes part of its signature and weight. In the new algebra, a path weighs less than another if the former weighs less than the latter in the primitive algebra or, the paths having the same weight in the primitive algebra, it comprises a smaller number of links. The new algebra is strict monotone, every network is free with respect to it, and so the path vector protocol converges. □

Proposition 4 does not prescribe any specific order for paths with the same origin, destination, weight, and number of links—any such order implies convergence.

Combining the necessity of monotonicity with Proposition 4 yields the following conclusion:

PROPOSITION 5. *The algebra is monotone if and only if there are relative path preferences for paths with the same weight that guarantee convergence of the path vector protocol in every network.*

From Proposition 5, we conclude that a path vector protocol converges to local-optimal-paths in-trees if and only if the algebra is monotone, and bearing on Proposition 2, that it converges to optimal-path in-trees if and only if the algebra is both monotone and isotone.

6.3 Checking convergence

In the previous section, we concluded that the convergence of path vector protocols hinges on the monotonicity and isotonicity of the underlying algebra and the freeness of the associated networks. In some cases, we will be able to exploit characteristics of the labels, signatures, and weights of the algebra to show those properties. In general, however, if there are $|L|$ labels and $|\Sigma|$ signatures, we need to perform $|L| \times (|\Sigma| - 1)$ compositions with the operation $\oplus$ and that same number of comparisons via the order $\preceq$ to verify monotonicity. As we do this, we should keep track, for every weight w, $w \neq f(\phi)$, of the set L_w of labels l for which there is at least one signature α such that $w = f(\alpha) = f(l \oplus \alpha)$. A free network is then a network where no cycle has links with labels taken exclusively from any one of the sets L_w. Verifying isotonicity, if needed, entails $|L| \times (|\Sigma| - 1) \times (|\Sigma| - 2)$ compositional operations and that same number of comparisons. By contrast, in combinatorial approaches the computational complexity of checking for convergence is a function of the number of possible paths in the network, which number is, in general, exponential in the size of the network.

7. EXAMPLES AND COUNTER-EXAMPLES

7.1 Roadmap

We now provide applications of the algebra. In Section 7.2, we deal with standard optimal path routing. Section 7.3 presents an example of an algebra that is monotone but not isotone. This is the composite metric of IGRP which, contrary to what one would expect, does not result in optimal path routing. Sections 7.4 and 7.5 formulate existing guidelines for policy-based routing with BGP in algebraic terms. These sections show that some guidelines comply with the concept of optimal paths, but more often, they only comply with the concept of local-optimal paths. Section 7.6 gives

Table 1: Example algebras for optimal path routing. We have $W = L = \Sigma$ and f is the identity mapping.

W	$\oplus$	ϕ	$\preceq$	Optimal path
$R_0^+ \cup \{+\infty\}$	$+$	$+\infty$	$\leq$	Shortest
$R_0^+ \cup \{+\infty\}$	min	0	$\geq$	Widest
$[0,1]$	$\times$	1	$\geq$	Most reliable
$\{(d,b) \mid d \in R_0^+, b \in R_0^+ \cup \{+\infty\}\} \cup \{\phi\}$	$(d_1 + d_2, \min(b_1, b_2))$	ϕ	$d_1 < d_2$ ou $d_1 = d_2$ e $b_1 \geq b_2$	Widest-shortest

an example of an algebra that is not monotone. Section 7.7 discusses performance-oriented extensions to BGP, and Section 7.8 gives alternative guidelines for policy-based routing. Last, in Section 7.9, the algebraic framework is used to derive a sufficient condition for signaling correctness of iBGP in domains that use route reflection.

7.2 Standard optimal paths

Table 1, borrowed from [15], presents instances of the algebra that are relevant to performance-oriented routing. In performance-oriented routing one is interested not only in the convergence of the routing protocol, but also on the quality of the paths the protocol has converged to. For all the examples of Table 1, the algebra is both monotone and isotone, so a path vector protocol can always be made to converge to optimal paths. The usual name of an optimal path is given in the last column. The first row corresponds to conventional shortest paths. The second, to widest paths. A widest path is a path of maximum width, where the width of a path is its capacity, which equals the capacity of its bottleneck link. The third row corresponds to most-reliable paths. The reliability of a path is the product of the non-failure probabilities of its constituent links. The fourth row corresponds to widest-shortest paths. A widest-shortest path is a widest path among the set of shortest paths from one node to another.

In the shortest path problem, a free network is a network in which every cycle has at least one link with length greater than zero. We can, for instance, conclude that if every link in a network has length greater than zero, then a path vector protocol always converges to shortest paths or to widest-shortest paths no matter the relative preferences given to paths with the same length or the same combination of length and width. In the widest path problem, every cycle makes a network non-free. In order for a path vector protocol to converge to widest paths, each node should prefer paths with the minimum number of links, among paths of the same width.

7.3 Non-isotonic algebra

IGRP [2] is a distance vector protocol and not a path vector protocol. We use its composite metric as an example of an algebra that is monotone but not isotone, against what one would expect to find in a performance-oriented environment. The conclusion that this composite metric does not make packets travel over optimal paths holds for both path vector and distance vector protocols.

In its most basic form, the composite metric of IGRP can be described by an algebra with $L = R^+ \times R^+$, $\Sigma = L \cup \{\epsilon, \phi\}$, $W = R_0^+ \cup \{+\infty\}$. The first component of a label represents length, and the second represents capacity. Accordingly, $(d_1, b_1) \oplus (d_2, b_2) = (d_1 + d_2, \min(b_1, b_2))$. The order $\preceq$ is $\leq$, and the function f is given by

$$f((d,b)) = d + \frac{k}{b},$$

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 $f((1,k/4) \oplus (2,k)) = f((3,k/4)) = 7 > 6 = f((2,k/4)) = f((1,k/4) \oplus (1,k/4))$.

7.4 Customer-provider and peer-peer relationships

We now turn to policy-based routing and BGP. In policy-based routing, the main goal is to make the path vector protocol converge. If and when it does converge, it converges to local-optimal paths, which may or may not be optimal paths.

The system in this section is taken from Guideline A in [4], and rests on the customer-provider and peer-peer relationships established between Internet domains [10]. We have $L = \{c, r, p\}$, $\Sigma = L \cup \{\epsilon, \phi\}$, and $W = \{0, 1, 2, +\infty\}$. The linear 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 will call *primary paths* to the usable paths obtained with the guidelines of this section. Primary paths are subdivided by 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; peer paths, whose first link is a peer link, have signature r; and provider paths, whose first link is a provider link, have signature p. The $\oplus$ operation 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 to become a customer path. In other words, a node does not export to a provider a path that it 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. Figure 4 depicts a network where links have

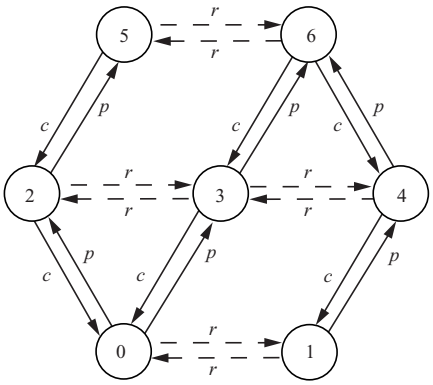

Figure 4: Network with customer-provider and peer-peer relationships. Labels are taken from the set $\{c, r, p\}$, where c, r, and p, identify customer, peer, and provider links, respectively. Peer links are represented with dashed lines as a visualization aid.

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

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

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. It turns out that this algebra is both monotone and isotone, so that the 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.

We use the procedure of Section 6.3 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 $1 = f(c) = f(c \oplus c)$; and $L_2 = \{p\}$, since $2 = f(r) = f(p \oplus r) = f(p) = f(p \oplus p)$. In conclusion, a free network is a network without cycles where all links have label c or all links have label p. 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. If we want to guarantee convergence of the path vector protocol without restricting the relationships between domains, it suffices to have each domain break ties within paths of same class, customer, provider, or peer, with the number of links in the path.

7.5 Backup paths

The system is taken from [3], and is an upgrowth of the system of the previous section that contemplates backup relationships between Internet domains. Backup relationships expand the set of usable paths to reach any particular destination, thus conferring robustness to the system in the presence of link failures. For example, if links $(6, 5)$ and $(3, 0)$ are down in the network of Figure 4, then the parsimonious relationships of the previous section would isolate node 6 from node 0. With the backup relationships of this section, node 6 could still reach node 0 over paths 6 3 2 0 and 6 4 1 0 for instance. We will call *backup paths* to the usable paths that are not primary paths. Every backup path contains at least one step as subpath. A *step* is a three-node path such that: the first link is a customer link and the second link is a peer link; both the first and the second links are peer links; or the first link is a peer link and the second link is a provider link.

We have $L = R^+ \times \{c, r\} \cup \{p\}$, $\Sigma = R_0^+ \times \{c, r, p\} \cup \{\epsilon, \phi\}$, and $W = R_0^+ \times \{1, 2\} \cup \{0, +\infty\}$. The set W is lexicographically ordered based on the order $\leq$. Trivial paths have signature ϵ. The signatures of non-trivial paths have two components. The first is called *avoidance level* and is such that the lower its value the most preferred the path. The second component is the class of the path, defined as in the previous section as a function of its first link: customer paths are marked with letter c; peer paths are marked with letter r; and provider paths are marked with letter p. As for labels, the letters c, r, and p identify customer, peer, and provider links, respectively. In a label of the form (y, c) or (y, r), the value y is positive and corresponds to the amount that the avoidance level of a path must increase when a step is found. The $\oplus$ operation is given in the next chart.

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

For example, $(y, c) \oplus (x, p) = \phi$ means that a node does not export a path learned from one provider to a different provider. In this system, this is the only restriction in exporting paths. As another example, $(y, c) \oplus (x, r) = (x+y, c)$ means that a customer can export a peer path to one of its providers, thus creating a step, but the avoidance level of the extended path must increase. In Figure 4, path 5 2 3 0 is a customer path containing step 5 2 3; path 0 3 2 5 is a provider path containing step 3 2 5; and path 4 3 2 0 is a peer path containing step 4 3 2. All these paths are backup paths. Path 2 0 3, for example, is neither primary nor backup, that is, it is not usable.

The function f is given next.

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

Note that the function f together with the order relation $\preceq$ gives predominance to the avoidance level of a path over its class, and that primary paths have an avoidance level of 0, meaning that they are always preferred to backup paths. The algebra is monotone but not isotone. The freeness condition is equivalent to the statement that there is no cycle where all links have labels taken from $R^+ \times \{c\}$, or all links have label p.

The system in [3] is more general than presented here in that the avoidance level of a path may also increase when there is no step, and the increase in avoidance level may

depend on properties of the path, other than its class. It is possible to account for the more general system with an expanded algebra.

7.6 Non-monotonic algebra

As an example of an algebra that is not monotone consider the algebra of the previous section but with the ordering of the set W being inverse-lexicographic, instead of lexicographic. That is, $(x_1, n_1) \preceq (x_2, n_2)$ if and only if $n_1 < n_2$, or $n_1 = n_2$ and $x_1 \leq x_2$. In this algebra, the class of the path has predominance over its avoidance level: a node always prefers customer paths to peer or provider paths; among customer paths, or among peer and provider paths, it prefers those with the smallest avoidance level. However, this algebra is not monotone, for $f((3, r)) = (3, 2) \succ (4, 1) = f((3 + 1, c)) = f((1, c) \oplus (3, r))$. With this algebra there are networks in which a path vector may never converge.

7.7 Performance extensions

There has been some interest in extending BGP to accommodate performance-aware parameters on top of policy guidelines [17]. Here, we take the simple case where the performance of a path is gauged only by its width to illustrate the general principle that compounding a monotonic algebra with another yields a monotonic algebra, but that compounding a isotonic algebra with another may not yield an isotonic algebra.

Let $(W', \preceq', L', \Sigma', \phi', \oplus', f')$ be the algebra that describes the policy guidelines of Section 7.4, and let $(W'', \geq, W'', W'', 0, \min, f'')$, with $W'' = R_0^+ \cup \{+\infty\}$ and f'' the identity mapping, be the algebra of widest paths (see Section 7.2). Both these algebras are isotonic. The compounded algebra that gives predominance to policy-based routing is the algebra with $W = W' \times W''$, $L = L' \times L''$, and $\Sigma = \Sigma' \times \Sigma''$. The $\oplus$ operation is given by $(\alpha_1, b_1) \oplus (\alpha_2, b_2) = (\alpha_1 \oplus' \alpha_2, \min(b_1, b_2))$, the function f is given by $f((\alpha, b)) = (f'(\alpha), b)$, and the order $\preceq$ is such that $(n_1, b_1) \preceq (n_2, b_2)$ if $n_1 < n_2$ or $n_1 = n_2$ and $b_1 \geq b_2$. This algebra is monotone but not isotone. For example, $f((c, 5)) = (1, 5) \prec (2, 10) = f((p, 10))$ whereas $f((p, 10) \oplus (c, 5)) = (2, 5) \succ (2, 10) = f((p, 10) \oplus (p, 10))$.

The practical conclusion to be taken from this discussion is that the performance-aware paths chosen by the nodes lack global significance in general: they are local-optimal paths, not optimal-paths. For instance, the provider path chosen by a node upon convergence of the protocol is not necessarily the widest among the provider paths that are usable at the node. In Figure 5, the numbers by the links represent their capacities. Upon convergence of the protocol, node 2 chooses path 2 1 to reach node 1, because it is the only customer path from node 2 to node 1, and customer paths are preferred to both peer and provider paths. This choice forces node 0 to choose provider path 0 2 1 to reach node 1. But provider path 0 2 1, of capacity 5, is not the widest provider path from node 0 to node 1: that distinction belongs to provider path 0 2 3 1 which has capacity 10.

7.8 Alternative guidelines

The approach described in Section 7.5 to include backup paths has two limitations. First, it allows valleys, if they cross peer links. A *valley* is a path that starts with a customer link and ends with a provider link, implying that a node may provide transit service to a provider. For example, with the algebra of Section 7.5, paths 3 0 1 4 and 4 1 0 3 would be allowed in the network of Figure 4, meaning that nodes 0 and 1 would provide transit service to their respective providers 3 and 4. Second, a node may prefer a backup path with provider links to one without provider links, because, as we have seen in Section 7.6, monotonicity would fail otherwise. For example, with the algebra of Section 7.5, and despite having a provider link, path 2 3 6 4 1 is preferred to path 2 0 1 if its avoidance level is lower.

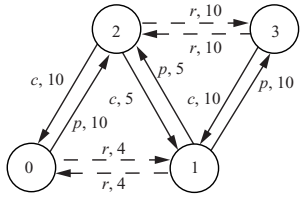

Figure 5: Network with customer-provider and peer-peer relationships. The numbers by the links represent their capacities.

We present alternative guidelines with the following four characteristics:

- 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 avoidance level of a backup path increases with every peer link that it contains.

We have $L = \{c, p\} \cup \{r\} \times R^+$, $\Sigma = \{\epsilon, c, p, \phi\} \cup \{r, \hat{c}, \hat{p}\} \times R^+$, and $W = \{0, 1, 2, 3, 4\} \times R_0^+ \cup \{+\infty\}$. The set W is lexicographically ordered based on the order $\leq$. In labels, the letters c, r, and p, again identify customer, peer, and provider links, respectively. The value 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. In signatures, the letters c, r, and p identify customer, peer, and provider paths, respectively, and the accented letters $\hat{c}$ and $\hat{p}$ identify backup paths without and with provider links, respectively. The value x in signatures of the form $(\hat{c}, x)$ and $(\hat{p}, x)$ indicates the avoidance level of a backup path. The signature of a peer path, of the form (r, x), inherits the value x from the label of its first link. Every trivial path has signature ϵ. The $\oplus$ operation is given next (the column for signature ϵ equals the one 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 customer link can never be prefixed to a path that contains provider links, 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 crosses a peer link.

The function f is given by

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

Note that in selecting a backup path, whether or not the path contains provider links takes precedence over its avoidance level. The freeness condition is equivalent to the statement that there is no cycle where all links have label c or all links have label p.

7.9 Route reflection

We now apply the concepts developed to study convergence of BGP inside an Internet domain (Autonomous System, AS) that employs route reflection [9, 8]. The routers inside an AS are partitioned into clusters. Each cluster contains a number of route reflectors, at least one, and their clients. For simplicity, we assume only one route reflector per cluster. 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. Given an IP prefix, external to the AS, the BGP route selection process prefers routes with the highest value of LOCAL-PREF attribute, and among these, it prefers routes with the lowest length of the AS-PATH attribute. We neglect the MED attribute, and from now on, we consider only the routes with highest LOCAL-PREF, and among these, only the ones with the lowest AS-PATH length. A router that learned at least one of these remaining routes from an eBGP (external BGP) session is called a *border router* for that IP prefix. The border routers are destinations as far as routing inside the AS is concerned.

We first identify the algebra that emerges from the route selection rules and export rules that are applied inside an AS that uses route reflection. The best of a set of available routes at a router is selected as follows: prefer the route with the shortest Interior Gateway Protocol (IGP) path distance to a border router, breaking ties with the identities of the border routers. The export rules are as follows: border routers export eBGP routes to all routers with which they have iBGP sessions; a route reflector exports routes learned from another route reflector only to all clients in its cluster; a route reflector exports routes learned from a client in the same cluster to all other clients in the cluster and all other route reflectors.

In the model, routers have identifiers taken from the set N of positive integers. We have $L = \{d, o, u\} \times N$, $\Sigma = (\{d, o\} \times N \times N) \cup (\{0, +\infty\} \times N) \cup \{\phi\}$, and $W = (R_0^+ \cup \{+\infty\}) \times (N \cup \{+\infty\})$, lexicographically ordered based on the order $\leq$. The second component in the label of each link is always the identity 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 $(0, k)$;

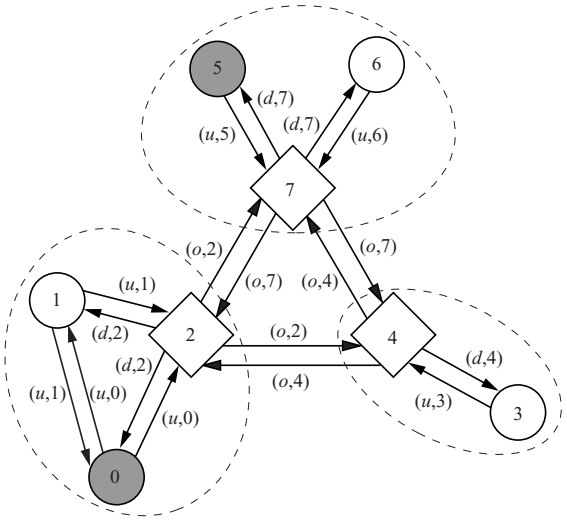

Figure 6: AS with three clusters. Clusters are enclosed in ovals. Route reflectors are represented with diamonds, clients are represented with circles, and border routers (for an unspecified IP prefix) are shaded.

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. Figure 6 depicts an AS that uses route reflection, and where the border routers, for an unspecified IP prefix, 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.

The $\oplus$ operation 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 k, which is an border router, 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.

$$f((0, k)) = (0, k)$$
$$f((d, i, k)) = f((o, i, k)) = (dist(i, k), k)$$
$$f((+\infty, k)) = (+\infty, k)$$
$$f(\phi) = (+\infty, +\infty)$$

where $dist(i, k)$ is the IGP path distance from router i to router k. With this algebra all networks are free, because the antecedent of the freeness condition is never true. We are left to verify monotonicity. 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 $dist(i,k) \leq dist(j,k)$.

We can then conclude with generality that the path vector protocol converges within an AS 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. The contrapositive states that for the path vector protocol not to converge within an AS at least one client must be closer to a router reflector other than the one in its cluster; examples of non-convergence can be found in [8].

8. PROOF OF CONVERGENCE

In this section, we present a semi-formal temporal-logic proof of Proposition 3. Specifically, we fix a destination and prove convergence of the protocol for that destination.

Let $\mathcal{P}$ be the set of all usable paths in the network through which the destination can be reached, and let the strict partial order $\lhd$ be defined such that $P \lhd Q$ if P and Q have the same origin and P weighs less that Q or, having the same weight as Q, is preferred to it. Define the *paths digraph* to be the digraph that has $\mathcal{P}$ for vertex set and where there is an edge from path P to path Q if any one of the next two conditions is verified:

- Q is an extension of P, that is, $Q = uv \circ P$ for some node u in the network;
- P and Q have the same origin, and either P weighs less than Q or, their weights being equal, P is preferred to Q, that is, $P \lhd Q$.

We remark that the use of the paths digraph is confined to the proof of Proposition 3, not being needed thereafter to prove convergence of specific path vector protocols. Figure 7 shows the paths digraph for the network of Figure 4, taking 0 for destination node. At the top part of the figure, the usable paths are depicted next to the nodes at their origin. The higher a path in a list the smaller it is with respect to the order $\lhd$. The bottom part of the figure shows the corresponding paths digraph.

PROPOSITION 6. *If the algebra is monotone and the network is free, then the paths digraph is acyclic.*

PROOF. The proof is by contradiction and comprises three stages. Assume that the paths digraph contains a cycle, and let $\mathcal{C} = P_0 \cdots P_{n-1} P_n$ $(P_n = P_0)$ be a cycle of minimum length. Since the paths usable at a node are totally ordered, and a path and any of its extensions have different origins, we must have $n \geq 4$. The origin of path P_i is denoted as u_i, $0 \leq i \leq n$.

In the first stage, we show that any repeated nodes in the sequence $u_0 \cdots u_{n-1} u_n$ $(u_n = u_0)$ must appear consecutively. Suppose otherwise. Then there is i, $0 \leq i < n$, and k, $1 < k < n-1$, such that $u_i = u_{i +_n k}$ and $P_i \lhd P_{i +_n k}$, where

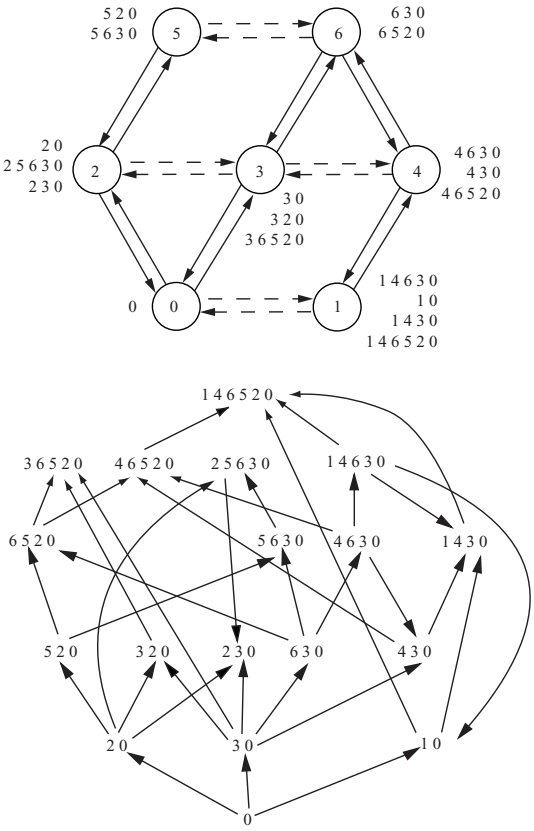

Figure 7: Example paths digraph for the customer-provider and peer-peer algebra of Section 7.4, and network of Figure 4.

$+_n$ denotes addition modulus n. If $i < i +_n k$, then the sequence $P_0 \cdots P_i P_{i +_n k} \cdots P_n$ is also a cycle in the paths digraph, and has length $n - k < n$, contradicting the minimality of $\mathcal{C}$. On the other hand, if $i +_n k < i$, then the sequence $P_{i +_n k} \cdots P_i P_{i +_n k}$ is a cycle in the paths digraph and has length $k+1 < n$, again contradicting the minimality of $\mathcal{C}$.

In the second stage, we show that there is $w \in W - \{f(\phi)\}$ such that $w = f(s(P_i))$ for $0 \leq i \leq n$. This follows easily from monotonicity. Let $w = f(s(P_0))$. Since P_0 is usable at node u_0, w is different from $f(\phi)$. Each edge of the paths digraph either joins a path to one of its extensions or joins a path to another that does not weigh less. Hence, $f(s(P_{i-1})) \preceq f(s(P_i))$, for $0 < i \leq n$. Because $P_n = P_0$, this set of inequalities can only be satisfied if $w = f(s(P_i))$, for $0 \leq i \leq n$.

In the third and final stage, we use freeness to arrive at the contradiction. First, we observe that the sequence of nodes obtained from $u_0 \cdots u_{n-1} u_n$ by skipping over repeated nodes and reversing their order is a cycle in the network. Formally, let m be the number of distinct nodes of $u_0 \cdots u_{n-1} u_n$. Define the function a from $\{0, \cdots, m\}$ to $\{0, \cdots, n\}$ as follows:

$$a(j) = \begin{cases} 0 & : \text{ if } j = 0 \\ a(j-1)+1 & : \text{ if } 1 < j \leq m \text{ and } u_{a(j-1)+1} \neq u_{a(j-1)} \\ a(j-1)+2 & : \text{ if } 1 < j \leq m \text{ and } u_{a(j-1)+1} = u_{a(j-1)}. \end{cases}$$

Because only consecutive nodes of $u_0 \cdots u_{n-1}u_n$ can be repeated, the sequence $u_{a(m)} \cdots u_{a(1)}u_{a(0)}$ $(u_{a(0)} = u_{a(m)})$ is a cycle in the network. Moreover, $P_{a(j)} = u_{a(j)}P_{a(j)-1}$, for $0 < j \leq m$. Letting $\alpha_i = s(P_i)$ and using the result from the second stage, we conclude that for every $0 < j \leq m$, we have $f(\alpha_{a(j)-1}) = w$ and $f(l(u_{a(j)}, u_{a(j)-1}) \oplus \alpha_{a(j)-1}) = w$, contradicting the freeness condition, and concluding the proof. □

We are now ready to present the semi-formal temporal-logic proof of Proposition 3. See [12] for background on temporal logics, and [11] for a temporal-logic proof of the convergence of conventional distance vector protocols.

Proposition 3: If the algebra is monotone and the network is free, then, whatever the relative preference given to paths with the same weight, the path vector protocol converges.

PROOF. Let $\mathcal{G}$ denote the paths digraph and d denote the trivial path consisting of the destination node alone. The *rank* of usable path P is defined to be one plus the number of edges in the longest path in $\mathcal{G}$ from vertex d to vertex P. In particular, the rank of path d is one. Because $\mathcal{G}$ is an acyclic graph, if there is an edge from P to Q, then Q is ranked higher than P. In other words, if Q is either an extension of P to some node, or has the same origin as P but more weight, or, having the same weight, is less preferred, then Q is ranked higher than P. The ranks of paths of $\mathcal{G}$ can be determined recursively, noting that the paths of rank j are those that have no edge pointing to them in the graph obtained from $\mathcal{G}$ by withdrawing all paths with rank less than j. Table 2 shows the ranking of the usable paths of Figure 7. Let M be the rank of the highest rank path in $\mathcal{G}$. We slightly abuse terminology to call *none* a path, to which we assign rank $M + 1$.

Suppose that the network topology has settled down. There will be a time when all state information related to paths that are not part of the network vanishes. We want to show that there will be a subsequent time when all signaling queues become empty: the path vector protocol will have converged when this happens. For this purpose, we 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 termination of the protocol [12, 11]. 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 routing messages that announce a path of rank j in transit in signaling queues **plus** number of nodes that have chosen a path of rank j.

Now, assume that a signaling routing message announcing path P, of rank j, arrives at node u coming from its out-neighbor v. Let Q, a path of rank k, be the chosen path at node u before the routing message was received, and let R, a path of rank l, be the chosen path at node u after the routing message is received. Four cases are distinguished.

1. $R = Q$: The coordinate f_j decreases by one. The function F decreases.

2. $R \neq Q$ and $R = uv \circ P$: The coordinate f_j decreases by one, the coordinate f_k decreases by one, and the coordinate f_l increases. Because R is the extension of P to node u, we have $l > j$ and, therefore, the function F decreases.

3. $R \neq Q$ and $R = none$: The coordinate f_j may decrease by one, the coordinate f_k decreases by one, and the coordinate f_{M+1} may increase. Because $R \neq Q$ and $R = none$, we have $k < M + 1$, and the function F decreases.

4. $R \neq Q$ and $R \neq none$ and $R \neq uv \circ P$: The coordinate f_j decreases by one, the coordinate f_k decreases by one, and the coordinate f_l increases. Because $R \neq uv \circ P$, path R was available for selection at node u before the signaling routing message was received. Since, in addition, $R \neq Q$ and path Q was the path chosen by u before the signaling routing message was received, Q weighs less than R or has the same weight as R but is preferred to it at node u. Hence, $k < l$ and the function F decreases.

□

9. ALGEBRA AND BGP

We now discuss the use of the algebraic framework in the design and implementation of policy guidelines for BGP. The ideas expressed in this section are preliminary and their merits need to be assessed by actual implementations. We view the algebraic framework as a mathematical template for setting up policy guidelines, expecting its semantic value to help in their translation to and from some router configuration language [5]. The set of labels corresponds to the class of possible types of relationships between pairs of nodes together with the relevant properties of the links joining them. The set of signatures reflects properties of paths that nodes are willing to keep and share with their neighbors. The mapping of signatures into weights and the translation of a signature to another via a label are such as to make the path vector protocol converge and satisfy any additional requirements that may be desired from the policy guidelines.

Once a set of policy guidelines is represented by an algebra, mapping its elements into BGP mechanisms comprises two main aspects. First, there must be some way to associate a signature with a BGP route. This can be achieved with the Community attribute. Second, we must be able to assign a weight to each Community value representing a signature. The set of weights can be created with the LOCAL-PREF attribute. However, if a weight corresponds to exactly one value of LOCAL-PREF, no margin is left for the ASes to individually apply routing policies that they do not wish to disclose to their neighboring ASes. A better solution is to let each weight correspond to a range of contiguous LOCAL-PREF values, with different weights having non-overlapping ranges. A router holding a route with a given Community value can choose any one of the LOCAL-PREF values associated with that Community value, the exact choice being outside the scope of the guidelines.

Routers need to be configured to respect the correspondence between Community values and ranges of LOCAL-PREF values and to convert among Community values representing signatures. External to the algebra is the decision of which router performs which conversions. For example, suppose that u and v are two routers in different ASes with

Table 2: Ranking of usable paths for the paths digraph of Figure 7.

rank 1	rank 2	rank 3	rank 4	rank 5	rank 6	rank 7	rank 8
0	2 0	3 2 0	4 6 3 0	4 3 0	1 0	1 4 3 0	1 4 6 5 2 0
	3 0	5 2 0	5 6 3 0	1 4 6 3 0	2 3 0		
		6 3 0	6 5 2 0	2 5 6 3 0	4 6 5 2 0		
				3 6 5 2 0			

a relationship described by label l, and that α is the Community value (signature) of a route hold by v. After the route is advertised to u its Community value becomes $l \oplus \alpha$. The algebra is the same whether the new Community value is computed at v before the route is advertised to u, or is computed at u after the route is received at u, or the computation is shared between u and v.

10. CONCLUSIONS

We have brought modern algebraic concepts to the design and study of routing strategies supported on path vector protocols. The convergence properties of path vector protocols are related with the monotonicity and isotonicity of the underlying algebra. Monotonicity is necessary and sufficient to make a path vector protocol converge, and monotonicity together with isotonicity are necessary and sufficient for convergence to optimal paths. We have also identified freeness as the property that a network must have for guaranteed convergence of path vector protocols independently of the relative preference given by the nodes to paths with the same weight.

The algebraic approach unites in a common framework previous results on optimal path routing and various guidelines for policy-based routing, and makes it easy to check the validity of new routing strategies. As examples of new applications, we have given guidelines for policy-based routing that contemplate backup relationships while rendering paths with valleys unusable, and we have derived a sufficient condition for convergence of iBGP in Internet domains that use route reflection. Last, we have used the framework to gain insight into the provision of QoS extensions to BGP.

As a final note, we remark that most of the theory developed here can be adapted to distance vector protocols, provided that we supplement these protocols with a mechanism to deal with the count-to-infinity problem. This is readily accomplished by endowing signaling routing messages with a counter that is incremented every time it its passed from one node to another. Limiting the value of this counter stops the counting to infinity.

11. ACKNOWLEDGEMENTS

I am grateful to Tim Griffin for the incitement to submit this work to SIGCOMM. I am also thankful to José Brázio, to Ramesh Govindan, my shepherd, and to the anonymous reviewers for the many comments that helped improve the paper.

12. REFERENCES

[1] M. Blanchet, F. Parent, and B. St-Arnaud. Optical BGP (OBGP): InterAS lightpath provisioning. Internet draft draft-parent-obgp-01.txt, January 2001.

[2] J. Doyle. *Routing TCP/IP*. Cisco Press, Indianapolis, IN, 1998. ISBN 1-57870-041-8.

[3] L. Gao, T. Griffin, and J. Rexford. Inherently safe backup routing with BGP. In *Proc. INFOCOM 2001*, pages 547–556, Anchorage, AK, April 2001.

[4] L. Gao and J. Rexford. Stable Internet routing without global coordination. *IEEE/ACM Transactions on Networking*, 9(6):681–692, December 2001.

[5] R. Govindan, C. Alaettinoglu, G. Eddy, D. Kessens, S. Kumar, and W. S. Lee. An architecture for stable, analyzable Internet routing. *IEEE Network*, 13:29–35, January/February 1999.

[6] T. Griffin, F. Shepherd, and G. Wilfong. Policy disputes in path-vector protocols. In *Proc. 7th International Conference on Network Protocols*, Toronto, Canada, November 1999.

[7] T. Griffin, F. Shepherd, and G. Wilfong. The stable paths problem and interdomain routing. *IEEE/ACM Transactions on Networking*, 10(2):232–243, April 2002.

[8] T. Griffin and G. Wilfong. On the correctness of IBGP configuration. In *Proc. SIGCOMM 2002*, pages 17–29, Pittsburgh, PA, August 2002.

[9] B. Halabi. *Internet Routing Architectures*. Cisco Press, Indianapolis, IN, 1997. ISBN 1-56205-652-2.

[10] G. Huston. Interconnections, peering and financial settlements. In *Proc. INET'99*, San Jose, CA, June 1999.

[11] L. Lamport. An assertional correctness proof of a distributed algorithm. *Science of Computer Programming*, 2(3):175–206, December 1982.

[12] L. Lamport. The temporal logic of actions. *ACM Trans. on Programming Languages and Systems*, 16(3):872–923, April 1994.

[13] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4). RFC 1771, March 1995.

[14] J. Rosenberg, H. Salma, and M. Squire. Telephony routing over IP (TRIP). RFC 3219, January 2002.

[15] J. L. Sobrinho. Algebra and algorithms for QoS path computation and hop-by-hop routing in the internet. *IEEE/ACM Transactions on Networking*, 10(4):541–550, August 2002.

[16] J. W. Stewart III. *BGP4: Inter-Domain Routing in the Internet*. Addison Wesley, Reading, MA, 1999. ISBN 0201379511.

[17] L. Xiao, K. S. Lui, J. Wang, and K. Nahrstedt. QoS extension to BGP. In *Proc. 10th International Conference on Network Protocols*, Paris, France, November 2002.