# Compact Policy Routing

Gábor Rétvári, András Gulyás, Zalán Heszberger, Márton Csernai, József J. Bíró
Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics
1117 Budapest, Magyar tudósok körútja 2. Hungary
Email: {retvari,gulyas,heszberger,csernai,biro}@tmit.bme.hu

July 29, 2011

**Abstract**

This paper takes a first step towards generalizing compact routing to arbitrary routing policies that favor a broader set of path attributes beyond path length. Using the formalism of routing algebras we identify the algebraic requirements for a routing policy to be realizable with sublinear size routing tables and we show that a wealth of practical policies can be classified by our results. By generalizing the notion of stretch, we also discover the algebraic validity of compact routing schemes considered so far and we show that there are routing policies for which one cannot expect sublinear scaling even if permitting arbitrary constant stretch.

## 1 Introduction

Compact routing theory is the research field aimed at identifying the fundamental scaling limits of shortest path routing and constructing algorithms that meet these limits [1–7]. Shortest path routing is a key ingredient in many modern network architectures, as it generally ensures low transmission delay while also minimizes the effort needed to transmit one unit of information from the source to the destination. To what extent shortest path routing can scale to large networks, in terms of the memory requirements of implementing the local forwarding functionality at network nodes, has for a long time been researched. It turns out that in general it is impossible to implement shortest path routing with routing tables whose size in all network topologies grows slower than linear with the increase of the network size [1, 2]. To answer this challenge, compact routing research seeks algorithms to decrease routing table sizes at the price of letting packets to flow along suboptimal

G. Rétvári was supported by the Janos Bolyai Fellowship of the Hungarian Academy of Sciences.

paths. In this context, suboptimal means that the forwarding paths are allowed to be longer than the shortest ones, but length increase is bounded by a constant stretch factor. By now, the research community has built a strong theoretical foundation for compact shortest path routing, fully characterizing its pinnacles and pitfalls on a broad catalog of network topologies including hypercubes, trees, scale-free networks, and planar graphs [4, 5, 8–10], while having defined efficient compact routing algorithms for the generic case as well [3, 4].

In order to ensure an expedient flow of information through the network, one often needs to provision routes taking into consideration a broader set of attributes beyond mere path length, such as path reliability and resilience constraints [11], bandwidth and perceived congestion [12–14], business relations and service level agreements between ISPs [15, 16], etc. These path selection strategies are usually described under the umbrella of policy routing. Practically speaking, a *routing policy* is a function that selects a preferred transmission route from the set of all forwarding paths available between two endpoints, according to predefined requirements. Indeed, a significant portion of the Internet today runs over policy routing [11, 12, 15, 17, 18]. Unfortunately, at the moment no theory is available to characterize the inherent scaling properties of these policy routing architectures, leaving a considerable gap in our understanding of their long term sustainability.

In this paper, we take the first steps towards filling this gap. We build on the recent work of Sobrinho and Griffin [19–22], who lay the theoretical foundations for describing disparate routing policy structures in a single theoretical framework using the notion of routing algebras, abstracting away their syntactic and semantic diversity and letting us to study them in a general, abstract sense. Using this framework, we give an algebraic characterization of the scalability of policy routing. As the main contribution of the paper, we determine the algebraic requirements for a policy to be implementable with sublinear routing tables and we give a comprehensive characterization of many practically important routing policies in networking. By generalizing the notion of stretch, we also explore the algebraic conditions under which the well-known shortest-path-based compact routing schemes [3, 4] generalize to policy routing and we show that introducing stretch cannot always eventuate sublinear scaling.

The rest of this paper is structured as follows. In Section 2, we introduce the basic notations and models used throughout the paper. Next, in Section 3 we characterize the local memory requirements for implementing an important subset of routing algebras, called regular algebras, and we apply the results to real-world routing policies. In Section 4 we deal with an algebraic interpretation of stretch and we generalize compact routing algorithms to regular algebras. Then, in Section 5 we discuss some practical considerations and finally Section 6 concludes the paper.

## 2 An algebraic model for policy routing

Let the communications network be modeled by a finite, connected, simple, undirected graph $G(V, E)$, let $|V| = n$ and let $|E| = m$. Communication between nodes is carried out by sending packets: neighboring nodes exchange packets directly, while remote nodes communicate through intermediate hops. We assume that nodes $v$ (edges $e$) are uniquely identified by a natural number $\mathrm{ID}(v)$ ($\mathrm{ID}(e)$). We often write simply $v$ ($e$) in place of $\mathrm{ID}(v)$ ($\mathrm{ID}(e)$). Let $\deg(v)$ denote the degree of $v \in V$ and let $d = \max_{v \in V} \deg(v)$. An $s - t$ *walk* is a sequence of nodes $p = (s = v_1, v_2, \ldots, v_k = t)$, where $k$ is the length of the walk and $(v_i, v_{i+1}) \in E : \forall i = 1, \ldots, k-1$, a *cycle* is a walk with $s = t$, and a *path* is a walk that visits a node at most once.

### 2.1 Routing algebras

Generally speaking, a routing policy can be considered as a function $p^*_{st} = \mathrm{Pol}(\mathcal{P}_{st})$ that selects from the set of available $s-t$ paths $\mathcal{P}_{st}$ a single *preferred* path $p^*_{st}$ according to some predefined rules. This definition is broad enough to contain basically every conceivable policy, including extreme cases like choosing a random path as well as traditional ones like shortest path routing.

To be more specific, we choose the abstract notion of *routing algebras* from Sobrinho and Griffin to describe routing policies within this paper [19–24]. This allows us to infer generic properties instead of having to define particular routing policies one by one and building piecemeal compact routing frameworks. In addition, it has been shown that basically all practically important routing policies possess an algebraic representation [21]. Thus, we shall use the terms routing policy and routing algebra interchangeably in this paper.

A routing algebra abstracts away the most important concepts of shortest path routing, namely weight composition (the method of constructing the weight of a path from the weights of its constituent edges) and weight comparison (expressing the preference between edges or paths). In this paper, a routing algebra $\mathcal{A}$ is defined as a totally ordered semigroup with a compatible infinity element: $\mathcal{A} = (W, \phi, \oplus, \preceq)$, where $W$ is the set of (abstract) weights that can be assigned to edges, $\phi$ ($\phi \notin W$) is a special infinity weight meaning that an edge/path is not traversable, $\oplus$ is a composition operator for weights, and $\preceq$ is weight comparison.

More formally, the following properties are presumed:

- $(W, \oplus)$ is a commutative semigroup
  - Closure: $w_1 \oplus w_2 \in W$ for all $w_1, w_2 \in W$
  - Associativity: $(w_1 \oplus w_2) \oplus w_3 = w_1 \oplus (w_2 \oplus w_3)$ for all $w_1, w_2, w_3 \in W$
  - Commutativity: $w_1 \oplus w_2 = w_2 \oplus w_1$ for all $w_1, w_2 \in W$

- $\preceq$ is a total order on $W$
  - Reflexivity: $w \preceq w$ for any $w \in W$
  - Anti-symmetry: if $w_1 \preceq w_2$ and $w_2 \preceq w_1$, then $w_2 = w_1$ for any $w_1, w_2 \in W$
  - Transitivity: if $w_1 \preceq w_2$ and $w_2 \preceq w_3$, then $w_1 \preceq w_3$ for any $w_1, w_2, w_3 \in W$
  - Totality: for all $w_1, w_2 \in W$ either $w_1 \preceq w_2$ or $w_2 \preceq w_1$
- $\phi$ is compatible with $(W, \oplus)$ according to $\preceq$
  - Absorptivity: $w \oplus \phi = \phi$ for all $w \in W$
  - Maximality: $w \prec \phi$ for all $w \in W$

Given a path $p = (v_1, v_2, \ldots, v_k)$ we obtain the weight $w(p)$ of $p$ by combining the weight of its constituent edges: $w(p) = \bigoplus_{i=1}^{k-1} w(v_i, v_{i+1})$. Then a *preferred* path in the algebra $\mathcal{A}$ between two nodes is simply one with the smallest weight according to $\preceq$:

$$\mathrm{Pol}(\mathcal{P}_{st}) = p^* : w(p^*) \preceq w(p), \forall p \in \mathcal{P}_{st} \ .$$

Now, one easily checks that shortest path routing corresponds to the algebra $(\mathbb{R}^+, \infty, +, \leq)$, while widest-path routing, where preferred paths are those with the largest bottleneck capacity, is simply $(\mathbb{R}^+, 0, \min, \geq)$. See further examples later in Section 3.1 and Section 5.

A special family of routing algebras, called *regular* routing algebras, will play an essential role in this paper.

**Definition 1.** *A routing algebra $\mathcal{A}$ is said to be* regular*, if it satisfies the following properties*[1]*:*

- *Monotonicity* (M)*: $w_1 \preceq w_2 \oplus w_1$ for all $w_1, w_2 \in W$*
- *Isotonicity* (I)*: $w_1 \preceq w_2 \Rightarrow w_3 \oplus w_1 \preceq w_3 \oplus w_2$ for all $w_1, w_2, w_3 \in W$*

Monotonicity (M) means that prepending an edge (or path) of weight $w_1$ with another edge (or path) of $w_2$ can only make it less preferred: $w_2 \oplus w_1 \succeq w_1$. By commutativity, the same applies to appending edges/paths: $w_1 \oplus w_2 \succeq w_1$. Isotonicity (I), on the other hand, requires $\preceq$ to be compatible with the semigroup $(W, \oplus)$ in the following sense: if an edge/path is preferred over some other one, then prepending or suffixing both with a common edge or path maintains this relation.

Below are some further algebraic properties we shall often use to characterize routing policies [22].

[1] In this paper, we use the definitions of Sobrinho [19] with the understanding that other authors may adopt different terminology. For instance, what will be called *isotonicity* here is called *monotonicity* in conventional order theory. The reason is that this terminology seems to be widely adopted in the literature.

- Delimited (D): $w_1 \oplus w_2 \neq \phi$ for all $\forall w_1, w_2 \in W$
- Strictly monotone (SM): $w_1 \prec w_2 \oplus w_1$ for all $w_1, w_2 \in W$.
- Selective (S): $w_1 \oplus w_2 \in \{w_1, w_2\}$ for each $w_1, w_2 \in W$.
- Cancellative (N): $w_1 \oplus w_2 = w_1 \oplus w_3 \Rightarrow w_2 = w_3$ for each $w_1, w_2, w_3 \in W$.
- Condensed (C): $w_1 \oplus w_2 = w_1 \oplus w_3$ for each $w_1, w_2, w_3 \in W$.

From the above, perhaps only delimitedness deserves more explanation. This property ensures that edges can be combined in an arbitrary sequence without the danger of obtaining an untraversable path. Intra-domain routing policies, like shortest path routing or widest path routing, are usually delimited, while inter-domain BGP routing policies are not.

## 2.2 Composite algebras

An attractive feature of routing algebras is that surprisingly complex and expressive policy constructions can be built using only an elemental set of primitive algebras by applying simple algebra composition and decomposition operators appropriately [21]. Two of these operators have particular importance in this paper, namely the lexicographic product operator [22] and subalgebras.

Given two routing algebras $\mathcal{A} = (W_{\mathcal{A}}, \phi_{\mathcal{A}}, \oplus_{\mathcal{A}}, \preceq_{\mathcal{A}})$ and $\mathcal{B} = (W_{\mathcal{B}}, \phi_{\mathcal{B}}, \oplus_{\mathcal{B}}, \preceq_{\mathcal{B}})$, the *lexicographic product* of $\mathcal{A}$ and $\mathcal{B}$ is a routing algebra $\mathcal{A} \times \mathcal{B} = (W, \phi, \oplus, \preceq)$ where

- $W = W_{\mathcal{A}} \times W_{\mathcal{B}}$, $\phi = (\phi_{\mathcal{A}}, \phi_{\mathcal{B}})$
- $(w_1, v_1) \oplus (w_2, v_2) = (w_1 \oplus_{\mathcal{A}} w_2, v_1 \oplus_{\mathcal{B}} v_2)$ for all $w_1, w_2 \in W_{\mathcal{A}}$ and $v_1, v_2 \in W_{\mathcal{B}}$
- $(w_1, v_1) \preceq (w_2, v_2) = \begin{cases} v_1 \preceq_{\mathcal{B}} v_2 & \text{if } w_1 =_{\mathcal{A}} w_2 \\ w_1 \preceq_{\mathcal{A}} w_2 & \text{otherwise} \end{cases}$

Note that $\phi$ is well-defined if $\mathcal{A}$ and $\mathcal{B}$ are delimited. In other cases, defining $\phi$ needs more attention.

**Proposition 1.** *The lexicographic product operator transforms the properties of the constituent algebras according to the following rules [22]:*

- $\mathrm{M}(\mathcal{A} \times \mathcal{B}) \Leftrightarrow \mathrm{SM}(\mathcal{A}) \vee (\mathrm{M}(\mathcal{A}) \wedge \mathrm{M}(\mathcal{B}))$
- $\mathrm{I}(\mathcal{A} \times \mathcal{B}) \Leftrightarrow \mathrm{I}(\mathcal{A}) \wedge \mathrm{I}(\mathcal{B}) \wedge (\mathrm{N}(\mathcal{A}) \vee \mathrm{C}(\mathcal{B}))$
- $\mathrm{SM}(\mathcal{A} \times \mathcal{B}) \Leftrightarrow \mathrm{SM}(\mathcal{A}) \vee (\mathrm{M}(\mathcal{A}) \wedge \mathrm{SM}(\mathcal{B}))$

The second algebra composition operator we consider in this paper is subalgebras. Given a routing algebra $\mathcal{A} = (W, \phi, \oplus, \preceq)$ and a weight set $W' \subseteq W$, the restriction of $\mathcal{A}$ to $W'$: $(W', \phi, \oplus, \preceq)$ is a subalgebra of $\mathcal{A}$ if and only if $W'$ is closed for $\oplus$. Subalgeras inherit the properties of the root algebra, but new ones may also emerge. For instance, the subalgebra $(\mathbb{R}^+, \infty, +, \leq)$ of the weakly monotone algebra $(\mathbb{R}^+ \cup \{0\}, \infty, +, \leq)$ is also strictly monotone.

### 2.3 Routing model

In order to describe the complex process of policy routing and forwarding, we generalize the model of *routing functions* from [1, 2]. In this model, a packet contains a payload plus a header[2] with routing related information. Now, given a routing policy $\mathcal{A}$ and a graph $G$, a *policy routing function* is a mapping $R : \mathbb{N} \times \mathbb{N} \mapsto \mathbb{N} \times \mathbb{N}$ together with a labeling of the nodes $L_V : V \mapsto \mathbb{N}$ and a labeling of the edges $L_E : E \mapsto \mathbb{N}$ with the following property: for every node pair $s, t$, the successive application of $R$

$$(h_{i+1}, l_{i+1}) = R(v_i, h_i), \ \ \forall i = 1, \dots, k-1$$

yields a preferred path $p^*_{st} = (s = v_1, \dots, v_i, \dots, v_k = t)$ according to $\mathcal{A}$ and corresponding edge labels $l_{i+1} = (v_i, v_{i+1})$, where $h_1$ is some appropriate initial header. We shall say that $R$ *implements* $\mathcal{A}$ on $G$ for indicating that $R$ produces preferred paths according to $\mathcal{A}$ on $G$.

Similarly to [1, 2], we assume that node labels (or addresses) can be encoded on $c \log n$ bits[3] for some $c$ constant. We further assume that for each node $v_i \in V$ the edges emanating from $v_i$ are labeled locally: $L_E(v_i, v_j) \in \{1, \dots, \deg(v_i)\}$. Additionally, the edge label $l_{i+1}$ is understood as coming from the local label space $L_E(v_i)$ of $v_i$. These limitations are to ensure that no extra routing information can be encoded into the labels besides pure identification. No such limitation exists, however, on the header size.

Now, routing according to the policy routing function $R$ occurs as follows. Upon receiving a packet with header $h$, a node $u$ simply evaluates its *local routing function* $R_u(h) = R(u, h)$ to obtain a new header $h'$ and an outgoing port at edge $l$. Then, $u$ sets the packet's header to $h'$ and forwards it on $l$. In general, this routing model is suitable to represent oblivious routing architectures, i.e., ones in which the route of a packet depends only on the contents of the packet itself and some static forwarding information. Yet, it is broad enough to capture basically any practically relevant forwarding scheme, like traditional destination-based and source-destination-based forwarding, label swapping, etc. For further details, consult [1, 2].

Introducing routing functions makes it comfortable to characterize the local memory needed at network nodes to implement a routing policy.

[2]Without loss of generality, headers can be represented by natural numbers.
[3]Logarithms are of base 2.

**Definition 2.** *The local memory requirement $M_{\mathcal{A}}$ of implementing the routing policy $\mathcal{A}$ is defined as:*

$$M_{\mathcal{A}} = \max_{G \in \mathcal{G}_n} \min_{R \in \mathcal{R}} \max_{u \in V} M_{\mathcal{A}}(R, u) \ ,$$

*where $M_{\mathcal{A}}(R, u)$ is the minimum number of bits needed to encode the local routing function $R_u$, $\mathcal{R}$ is the set of all policy routing functions implementing $\mathcal{A}$ on some graph $G$, and $\mathcal{G}_n$ is the set of all graphs of size $n$.*

A routing policy is said to be *incompressible*, if $M_{\mathcal{A}}$ is $\Omega(n)$. Otherwise $\mathcal{A}$ is *compressible*. Easily, an incompressible routing policy does not scale well, as the memory needed to store the local routing process of some node increases with the number of nodes in at least one graph. On the other hand, compressible routing policies scale well.

### 2.4 Algebraic compact routing

At this point, we have all the definitions in place to focus on our main concern what we call algebraic compact routing: given a routing algebra describing a particular routing policy, *(i)* identify the theoretical bounds on the memory requirements needed to implement that algebra and *(ii)* examine the local storage vs. path optimality trade-off, that is, design compact routing schemes that implement the algebra with sublinear local storage at the price of letting traffic to flow along non-preferred paths, whose suboptimality is upper bounded by some suitably defined stretch.

From the standpoint of routing, regular algebras manifest the "well-behaved cases" [19, 20, 23]. Monotonicity and isotonicity on the one hand guarantee that the preferred paths themselves can be obtained in polynomial time using a generalization of Dijkstra's algorithm. On the other hand, in a regular algebra preferred paths emanating from a node always make up a tree, allowing for a single routing entry to be maintained with respect to each node and forwarding packets based on the destination address only. This allows us to store local routing information on at most $\tilde{O}(n)$ bits local memory. We formulate these ideas as follows.

For some graph $G$ and algebra $\mathcal{A}$, define a *destination-based routing function* $\hat{R}$ for implementing $\mathcal{A}$ on $G$ as follows. Let the packet header consist of the identifier of the packet's destination and let node $u$ forward a packet destined to some $v$ on the first edge $l_v$ along the preferred path $p^*_{uv}$: $\hat{R}_u(v) = (v, l_v)$. Sobrinho makes the following observation [20]:

**Proposition 2.** *$\mathcal{A}$ can be implemented by a destination-based routing function on any graph, if and only if $\mathcal{A}$ is regular.*

One easily sees that $\hat{R}$ basically corresponds to destination-oriented routing tables, storing a single entry for each destination node. This leads to the following observation.

**Observation 1.** *If $\mathcal{A}$ is regular, then it can be implemented using $O(n \log d)$ bits local information.*

A key question in compact routing research is whether this trivial routing function is optimal in the sense that it requires the minimum possible local memory to encode preferred paths, or there are better algorithms using less local space. For shortest path routing in particular, Fraigniaud and Gavoille present the following negative result [1, 2].

**Proposition 3.** *The shortest path routing algebra $\mathcal{A} = (\mathbb{R}^+, \infty, +, \leq)$ is incompressible.*

For shortest path routing at least, routing tables are optimal. For other routing policies, no such results exist. Therefore, in the next section we give an algebraic characterization of the memory requirements of policy routing.

## 3 Local memory requirements of policy routing

In what follows, we discuss the algebraic requirements for a routing policy to be implementable with sublinear local storage and we also give negative results indicating incompressibility of some practically important routing policies.

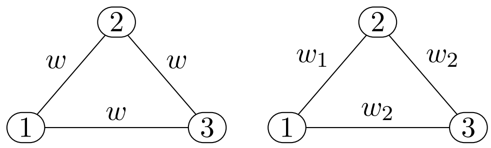
**Theorem 1.** *If $\mathcal{A}$ is selective and monotone, then it is compressible.*

In fact, we shall prove a bit more. We shall show that if a routing policy is selective, then a "preferred" spanning tree always exists, that is, for any $s, t \in V$ the only path $p_{st}$ contained in the tree is a preferred path. We say that algebra $\mathcal{A}$ *maps to a tree*, if for any connected graph and any weighing of the edges one can always find such a "preferred" spanning tree. Then, compressibility follows as routing over a tree is possible with $\log n$ bits local memory.

**Lemma 1.** *$\mathcal{A}$ maps to a tree, if and only if $\mathcal{A}$ is selective and monotone.*

*Proof.* To prove sufficiency, we construct an optimal spanning tree assuming that the algebra is selective and monotone. Take the edges in order of non-decreasing weight according to $\preceq$, add an edge to the spanning tree $T$ if no cycle arises and terminate when $T$ spans $G$. We show that the only in-tree path $p_{st}^T$ between any two nodes $s$ and $t$ is a preferred path over $\mathcal{A}$. To see this, take any other $s - t$ path $p_{st}$ in $G$. Obviously, there is at least one edge $(u, v)$ in $p_{st}$ so that $w(u, v) \succeq w(i, j)$ for all $(i, j)$ in $p_{st}^T$. Then, due to selectivity $w(p_{st}^T) \in \{w(i, j) : (i, j) \text{ in } p_{st}^T\}$, and by monotonicity $w(p_{st}^T) \preceq w(u, v) \preceq w(p_{st})$, therefore $p_{st}^T$ is a preferred $s - t$ path. This proves sufficiency.

Next, we prove that if $\mathcal{A}$ maps to a tree then $\mathcal{A}$ is monotone and selective. Easily, $\mathcal{A}$ is monotone, otherwise preferred paths might contain loops. Next,



(a) $w \oplus w \succ w$ (b) $w_1 \oplus w_2 \succ w_2$

Figure 1: Counter-examples for different violations of selectivity.

we show that if $\mathcal{A}$ is non-selective, then in some graphs preferred paths do not reside in a tree. Obviously, a monotone non-selective algebra $\mathcal{A}$ either contains a weight $w \in W$, so that $w \oplus w \succ w$ (auto-selectivity), or $\mathcal{A}$ contains two weights $w_1, w_2 \in W, w_1 \prec w_2$, so that $w_1 \oplus w_2 \succ w_2$. For both of these cases, Fig. 1 gives counter-examples in which the preferred paths are always through the direct edges, and so preferred paths do not make up a tree. Thus, for any non-selective algebra there is a graph in which preferred paths are not in a tree, which concludes the proof. □

A special case of this result for minimum- and maximum-type of weight composition operators appeared in [25], and [24] gives similar results for special routing algebras called *dioids*.

Theorem 1 suggests that routing policies characterized by selective algebras can be implemented using tree routing schemes, needing only logarithmic sized local storage [4, 10]. In contrast to selective algebras however, many routing policies can only be implemented using at least $\Omega(n)$ bits local memory as the next result shows.

**Theorem 2.** *If $\mathcal{A}$ is strictly monotone, then it is incompressible.*

We shall prove a deeper, more general claim, of which the above is a simple corollary.

**Lemma 2.** *If $\mathcal{A}$ contains a delimited, strictly monotone subalgebra, then $\mathcal{A}$ is incompressible.*

Table 1: Local memory requirements of various routing policies.

| Algebra | Definition | Properties | Local memory |
|---|---|---|---|
| Shortest path | $\mathcal{S} = (\mathbb{R}^+, \infty, +, \leq)$ | SM, I | $\Theta(n)$ |
| Widest path | $\mathcal{W} = (\mathbb{R}^+, 0, \min, \geq)$ | S, I, M | $\Theta(\log n)$ |
| Most reliable path | $\mathcal{R} = ((0, 1], 0, *, \geq)$ | SM, I | $\Theta(n)$ |
| Usable path | $\mathcal{U} = (\{1\}, 0, *, \geq)$ | S, I, M | $\Theta(\log n)$ |
| Widest-shortest path | $\mathcal{WS} = \mathcal{S} \times \mathcal{W}$ | SM, I | $\Theta(n)$ |
| Shortest-widest path | $\mathcal{SW} = \mathcal{W} \times \mathcal{S}$ | SM, $\neg I$ | $\Omega(n)$ |

*Proof.* We trace back incompressibility to the incompressibility of shortest path routing (Proposition 3), by showing that a delimited, strictly monotone algebra has subalgebras possessing the same structure as shortest path routing. We use the following basic facts from semigroup theory [26]. Every element $w \in W$ of a semigroup $(W, \oplus)$ generates a subsemigroup, the so called cyclic semigroup, $(W_w, \oplus) : W_w = \{w, w^2, w^3, \ldots\}$ through the power operation:

$$\forall n \in \mathbb{N} : \quad w^n = \begin{cases} w & \text{if } n = 1 \\ w \oplus w^{n-1} & \text{otherwise} \end{cases}$$

If the ordered semigroup $(W, \oplus, \preceq)$ is delimited and strictly monotone, then any of its cyclic subsemigroups $(W_w, \oplus)$ is of infinite order, in which case it is isomorphic to the semigroup $(\mathbb{N}, +)$ of natural numbers under addition through the mapping $f : \mathbb{N} \leftrightarrow W_w,\ f(n) = w^n$. In addition, $f$ is also an order preserving isomorphism between the shortest path routing algebra $S = (\mathbb{N}, \infty, +, \leq)$ and $(W_w, \phi, \oplus, \preceq)$ in this case, as $i < j \Leftrightarrow w^i \prec w^j$ due to strict monotonicity. One easily checks this by observing that for any $i < j : w^i \prec w^i \oplus w = w^{i+1} \preceq w^j$. Thus, if $\mathcal{A} = (W, \phi, \oplus, \preceq)$ has a strictly monotone subalgebra, then for any graph $G$ and any labeling of the edges of $G$ by natural numbers as weights, we can construct a labeling using weights from $W$ so that a path is a shortest path in the algebra $\mathcal{S} = (\mathbb{N}, \infty, +, \leq)$ if and only if it is a preferred path in $\mathcal{A}$. This implies that routing in $\mathcal{A}$ requires at least as much local memory as shortest path routing (i.e., $\Omega(n)$ by Proposition 3), which completes the proof. □

## 3.1 Examples

In Table 1, we list the intra-domain routing policies studied most extensively in the literature, together with their algebraic definition, basic properties, and the local memory requirements as indicated by our results. Note that all the listed algebras are delimited and regular except the last one. Here, $\mathcal{S}$ is the well-known shortest path routing algebra, for which Proposition 3 provides an adequate incompressibility characterization. Easily, Theorem 2 gives the same characterization.

$\mathcal{W}$ denotes the widest path routing policy [12]. Here, the weight of an edge is its capacity, the end-to-end capacity of a path equals the bandwidth of its bottleneck edge (the one with the smallest capacity) and the higher the capacity along a path the more preferred. Easily, this corresponds to the selective algebra $(\mathbb{R}^+, 0, \min, \geq)$, and so $\mathcal{W}$ is compressible by Theorem 1. In particular, under the tree routing scheme due to Fraigniaud and Gavoille [10] widest path routing can be implemented using $5 \log n$ bit addresses and $3 \log n$ bits local memory, or $\log^2 n$ bits using the scheme of Thorup and Zwick [4]. Similar is the case for the usable path routing strategy ($\mathcal{U}$), applied extensively in Ethernet switching[4]. However, the rest of the routing policies listed in the table are incompressible.

Most reliable path routing ($\mathcal{R}$) denotes the policy when edges are assigned a reliability metric denoting the possibility that a packet will be transmitted successfully over the edge and the path with the highest probability of success is favored. Easily, $\mathcal{R}$ contains a strictly monotone subalgebra. Widest-shortest path ($\mathcal{WS}$) routing prefers from the set of shortest paths the one with the highest free capacity [13], and shortest-widest path ($\mathcal{SW}$, [12, 14]), just contrarily, prefers the shortest one out of the set of widest paths. These algebras can be expressed as lexicographic products of the $\mathcal{S}$ and $\mathcal{W}$ algebras and, by Proposition 1, strictly monotone [22]. Hence, for $\mathcal{R}$ and $\mathcal{WS}$, which are isotone, Theorem 2 supplies the local memory requirement of $\Omega(n)$. This characterization is tight apart from a logarithmic factor, as simple table-based destination-oriented routing requires $\widetilde{\mathrm{O}}(n)$ bits by Observation 1. On the other hand, $\mathcal{SW}$ is not isotone. Theorem 2 holds for non-isotone algebras as well, which supplies a $\Omega(n)$ bits local memory requirement for $\mathcal{SW}$ too. At the moment, it is an open question whether this characterization is tight, as the only trivial routing function for $\mathcal{SW}$ stores a separate routing table entry for each source-destination pair, which needs $O(n^2 \log d)$ bits per router.

# 4 Compact policy routing

As has been shown in the previous section, many practically relevant routing policies are impossible to implement with sublinear size routing tables. In the case of shortest path routing, a standard way to improve scalability is to define *compact routing schemes*. In these schemes, paths are allowed to be longer than the shortest one, but path increase is upper bounded by a *multiplicative stretch factor* $k$, meaning that the paths yielded by the compact routing scheme are at most $k$ times as long as the shortest one. In the followings, we characterize the routing policies that admit similar compact implementations, at least for a sufficient abstract notion of stretch. Consider the following definition:

**Definition 3.** *A routing scheme is of stretch $k$ over algebra $\mathcal{A}$, if for any path $p_{st}$ selected by the scheme: $w(p_{st}) \preceq (w(p^*_{st}))^k$, where $p^*_{st}$ is some preferred $s-t$ path in $\mathcal{A}$.*

Note that $(w(p^*_{st}))^k = \underbrace{w(p^*_{st}) \oplus w(p^*_{st}) \cdots \oplus w(p^*_{st})}_{\text{k times}}$, which implies that the above definition indeed generalizes the notion of multiplicative stretch originally defined for shortest path routing.

[4] The fact that Ethernet runs over what is called the Spanning Tree Protocol shows the expressiveness of Lemma 1.

## 4.1 Algebraic requirements of compact policy routing

First, we ask which routing algebras lend themselves to be implemented by a compact routing scheme of finite stretch.

**Theorem 3.** *If a routing algebra $\mathcal{A}$ is regular, then there is a stretch-3 compact routing scheme for $\mathcal{A}$.*

We show that the stretch-3 shortest path routing scheme due to Cowen [3] readily generalizes to regular algebras. Below, we briefly reproduce that scheme. For further details, see [3] and [4].

For each $u \in V$, choose some node set $L \subseteq V$ and with each $u \in V$ associate a *landmark* $l_u$ as the node closest (according to $\mathcal{A}$) to $u$ in the set $L$. Additionally, for each $u \in V$ define a *ball* $B(u) : \{v \in V : w(p^*_{u,v}) \preceq w(p^*_{u,l_u})\}$, where $p^*_{s,t}$ refers to the preferred $s-t$ path for any $s$ and $t$. Finally, let the *cluster* of $u$ be $C(u) = \{v \in V : u \in B(v)\}$. When $\mathcal{A}$ is regular, one can use the lexicographic lightest path algorithms in [19,20] to obtain unique connected clusters for each $u$.

The routing scheme is a hop-by-hop technique. The label of node $v$ consists of the triplet $(v, l_v, \mathrm{port}_{l_v,v})$, where $v$ is the identifier of the node, $l_v$ is the identifier of its corresponding landmark, and $\mathrm{port}_{l_v,v}$ is the local port at $l_v$ to the first hop on the preferred path from $l_v$ to $v$. The packet header is the label of the target node. The routing table at node $u \notin L$ consists of $(v, \mathrm{port}_{u,v})$ tuples with respect to each $v \in C(u) \cup L$, where $\mathrm{port}_{u,v}$ is again the local port label of the first edge along the preferred $u - v$ path.

Packet forwarding *inside* a cluster occurs along preferred paths using the entries in the local routing tables. To route a packet to a node $v$ *outside* the cluster, node $u$ first forwards the packet to $v$'s landmark, from where it arrives to $v$ using again a direct route. In particular, when a packet with target $v$ arrives to a node $u \neq v$, $u$ checks whether $v$ is contained in its local routing table. If not, then $l_v$, the landmark of $v$ is extracted from the header. If $u = l_v$, then appropriate port label is also extracted from the header, otherwise it is looked up in the local routing table. Forwarding terminates when $u = v$.

From Proposition 2, we know that if $\mathcal{A}$ is regular, then standard destination-based hop-by-hop routing is correct. To show that the above scheme is also correct, the following crucial fact is enough (observed for shortest path routing by Cowen in [3]).

**Lemma 3.** *Suppose that $\mathcal{A}$ is monotone. Now, if $u$ stores an entry in its local routing table towards some $t$, then the next hop $v$ along the preferred $p^*_{ut}$ path also stores an entry to $t$.*

*Proof.* Easily, by monotonicity $p^*_{vt} \preceq p^*_{ut} \preceq p^*_{l_t,t}$ so $v$ also stores an entry for $t$. □

Next, we show that the scheme is stretch-3 on $\mathcal{A}$. As forwarding inside clusters occurs along preferred paths, we only need to prove stretch-3 for indirect forwarding via landmarks.

**Lemma 4.** *If $\mathcal{A}$ is regular, then for any $u, v \in V$ with $v \notin C(u)$ : $w(p^*_{u,l_v}) \oplus w(p^*_{l_v,v}) \preceq (w(p^*_{u,v}))^3$.*

*Proof.* *(i)* by assumption, $w(p^*_{l_v,v}) \preceq w(p^*_{u,v})$; *(ii)* using the triangle inequality, $w(p^*_{u,l_v}) \preceq w(p^*_{u,v}) \oplus w(p^*_{v,l_v}) = w(p^*_{u,v}) \oplus w(p^*_{l_v,v})$ (the latter equlality comes by commutativity); *(iii)* by isotonicity, from *(i)* and *(ii)* we have $w(p^*_{u,l_v}) \preceq w(p^*_{u,v}) \oplus w(p^*_{u,v})$. Combining *(i)* and *(iii)* by isotonicity we obtain $w(p^*_{u,l_v}) \oplus w(p^*_{l_v,v}) \preceq w(p^*_{u,v}) \oplus w(p^*_{u,v}) \oplus w(p^*_{u,v})$. □

Finally, we show that the local information is indeed sublinear. Obviously, addresses can be encoded on $3 \log n$ bits. The size of the local routing table at node $u$ is $O(|C(u)| + |L|)$. Using the landmark selection technique given by Cowen one obtains a local memory requirement of $O(n^{2/3})$ [3], which is improved by Thorup and Zwick to $\widetilde{\mathrm{O}}(n^{1/2})$ in [4].

An extremely interesting case is when the policy is the widest-path routing algebra $\mathcal{W}$. In this case, for any $n \in \mathbb{N}$ and any $w \in W : w^n = w$. Hence, stretch-3 paths are exactly the preferred paths in this case. The same applies to any selective and monotone algebra. Thus, Theorem 3 in fact gives an alternative proof to the claim that monotone and selective algebras are compressible.

We argued in Section 2.4 that regular algebras are the "well behaved" cases from the aspect of distributed routing, as they can be implemented by destination-based routing tables. Our results so far indicate that regular algebras are "well-behaved" from the standpoint compact routing as well: not just that we could give a general result characterizing the memory requirements for implementing regular algebras, but we also found that even when a regular algebra turns out incompressible a stretch-3 compact routing scheme is guaranteed to exist. In the next section, we show that if regularity fails, then finite stretch compact routing becomes significantly more difficult.

### 4.2 Compact routing when isotonicity fails

We have shown that regularity of a routing algebra is sufficient to define a stretch-3 compact routing scheme. It is an intriguing question whether it is necessary as well. At the moment, we do not have an answer to this question. What we can show, however, is that when isotonicity fails in a very intricate way, then no stretch-$k$ routing exists for any $k$ constant.

**Theorem 4.** *Let $k \geq 1$ and let $\mathcal{A} = (W, \phi, \oplus, \preceq)$ be a monotone algebra with the property that for any $p \geq 2$, $\exists\{w_1, w_2, \ldots, w_p\} \subseteq W$ so that $\forall i, j \in \{1, \ldots, p\}, i \neq j$:*

$$w_i \oplus w_j \succ w_i^{2k} \text{ and } w_i \oplus w_j \succ w_j^{2k} \quad . \tag{1}$$

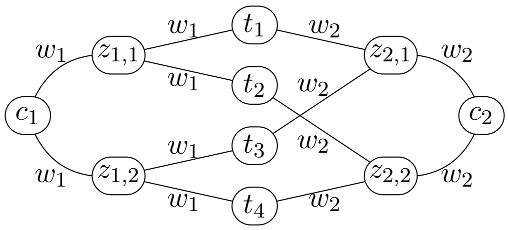



Figure 2: A sample graph for $p = 2$, $\delta = 2$ if the words for the target nodes are $[1,1]$, $[1,2]$, $[2,1]$ and $[2,2]$.

*Then, there is no stretch-$k$ routing scheme with sublinear memory requirement at all nodes.*

*Proof.* Borrowing the idea from [1], we present a family of graphs on which any stretch-$k$ implementation of $\mathcal{A}$ requires $\Omega(n)$ bits at some nodes. Start with a set of nodes $c_i \in C$, $|C| = p \geq 2$. To each $c_i \in C$, add $\delta \geq 2$ neighbors $z_{ij}, i \in \{1, \dots, p\}, j \in \{1, \dots, \delta\}$ and label the edges by $w_i$. Finally, add $\delta^p$ nodes $t \in T$ and connect these to the $z_{ij}$ nodes according to the following rule: for each $t \in T$ take the alphabet consisting of the symbols $(1, \dots, \delta)$, construct a word of length $p$ from this alphabet and add an edge from $z_{ij}$ to $t$ if the $i$th symbol in the word is exactly $j$. Label any $(z_{ij}, t)$ edge by $w_i$. Fig. 2 gives an example.

By monotonicity and (1), the preferred path $p^*_{c_i,t}$ from any $c_i \in C$ to any $t \in T$ is the min-hop path, so $w(p^*_{c_i,t}) = w_i \oplus w_i = w_i^2$. Fraigniaud and Gavoille in [1] show that encoding these paths in the above family of graphs requires $\Omega(n \log \delta)$ bits of storage space at the nodes in $C$. Intuitively speaking, the idea is that there is an astronomical number of different graphs in this graph family, and to encode the min-hop paths the routing algorithm needs to be able to differentiate amongst them, which requires huge storage space.

Unfortunately, any stretch-$k$ compact routing scheme for $k$ finite needs to encode the exact same min-hop paths. By construction, any non-preferred path $p_{c_i,t}$ goes through at least two edges of weight $w_j$ for some $j \in \{1, \dots, p\}, j \neq i$, and hence is at least of stretch $k$: $w(p_{c_i,t}) \succeq w_i \oplus w_i \oplus w_j \oplus w_j \overset{\text{(i)}}{=} (w_i \oplus w_j) \oplus (w_i \oplus w_j) \overset{\text{(ii)}}{\succeq} w_i \oplus w_j \overset{\text{(iii)}}{\succ} (w_i^2)^k = w(p^*_{c_i,t})$, where *(i)* is by associativity and commutativity, *(ii)* is by monotonicity, and *(iii)* is by (1). $\square$

A key to the above result is the weight set with the special structure (1), an extreme form of strict monotonicity. For $k \geq 2$, (1) violates isotonicity, therefore the theorem does not apply to regular algebras. But to many non-regular algebras it does. For the shortest-widest path policy in particular, one easily generates the weights $w_i$ with the required properties. Let $w_i = (b_i, c_i)$, where $b_i$ denotes the capacity and $c_i$ a positive cost, and for each $i = 1, \dots, p$

Table 2: Weight composition in valley-free routing.

| $\oplus$ | $c$ | $r$ | $p$ |
|---|---|---|---|
| $c$ | $c$ | $\phi$ | $\phi$ |
| $r$ | $r$ | $\phi$ | $\phi$ |
| $p$ | $p$ | $p$ | $p$ |

choose $b_i = i$ and let $c_i = (2k)^{i-1}$. One easily checks that this construction satisfies (1), since if $i < j$ then $b_i < b_j$ implies $(b_i, c_i)+(b_j, c_j) = (b_i, c_i+c_j) > (b_j, c_j)^{2k}$, while from $c_i < 2kc_i \leq c_j$ we get $(b_i, c_i + c_j) > (b_i, c_i)^{2k}$. This then implies that the shortest-widest path policy does not admit a compact implementation by Theorem 4.

# 5 Practical implications

We have seen that regular algebras are the easy cases for compact policy routing. However, many real-world routing policies do not lead to regular algebras (or commutative, or associative algebras, for that matter). The most prominent of these is the routing policies used by the Border Gateway Protocol (BGP), the inter-domain routing mechanism that glues the Internet together [27, 28]. Below, we very briefly discuss to what extent the above algebraic treatment can be applied to BGP policy routing algebras.

BGP policy routing can be described at various levels of depth. At the first, elemental level, BGP policy routing corresponds to the *valley-free routing* policy: each edge is labeled as *customer* ($c$), *peer* ($r$) or *provider* ($p$), and the only rule is that no path can contain a $c - p$, $c - r$, $r - p$, or a $r - r$ subpath [29]. This policy can be described by the algebra $\mathcal{B}_1 = (\{p, r, c\}, \phi, \oplus, \preceq)$, where $\oplus$ given in Table 2 and all permitted paths (i.e., ones whose weight is not $\phi$) have the same preference [20, 21]. To correctly represent the valley-free routing policy, the underlying graph is supposed to be a digraph in which the opposite arc of a $p$ ($r$) arc is always labeled as $c$ ($r$, respectively). Furthermore, $\oplus$ is right-associative. In line with what we see in the Internet, it is usually assumed that every node has a valley-free route to every other node and the network contains no provider-loops (directed $p$-cycles). Even though this setting violates basically every assumption in terms of which we stated our previous results, the basic ideas are still applicable as illustrated below.

**Theorem 5.** *$\mathcal{B}_1$ is compressible.*

*(Sketch).* By temporarily neglecting peer arcs, split the graph to *strongly connected valley-free components* (SVFC) with the property that in each component any pair of nodes $u, v$ can be bidirectionally connected by a valley-free path using customer-provider arcs only. In each SVFC, valley-free routing reduces to the selective and monotone subalgebra $\mathcal{B}'_1 = (\{p\}, \phi, \oplus, \preceq)$

with $p \oplus p = p$. As the graph contains no provider loops, every SVFC has a single node, call this the root node, that possesses no outgoing provider link. Then, a straightforward extension of Lemma 1 yields that routing inside a SVFC according to $\mathcal{B}_1'$ equals routing on an arborescence, which is possible with $O(\log n)$ local memory. Furthermore, roots are connected in a full-mesh due to global reachability, routing on which can be done using $O(\log n)$ local memory by a special port labeling [30]. The combination of these two routing schemes yields an $O(\log n)$ routing scheme for valley free routing. $\square$

At the second level, BGP classifies paths according to the *local preference* rules. A minimalistic rule contained in basically every local preference setting is that customer paths are favored over peer and provider paths. This can be described by the algebra $\mathcal{B}_2 = (\{p, r, c\}, \phi, \oplus, \preceq)$, where $\oplus$ again is as in Table 2 and $c \prec r \preceq p$.

**Theorem 6.** *$\mathcal{B}_2$ is incompressible. Additionally, there is no stretch-$k$ compact routing scheme for $\mathcal{B}_2$ for any finite $k \geq 2$.*

*(Sketch).* We show a weight set satisfying (1), from which Theorem 4 gives the required result. Simply, let $w_i = c$. As customer arcs are exactly provider arcs in the reverse direction, we have that the weight of any non-preferred path is at least $c \oplus p = \phi \succ c^k$ for any $k \geq 1$. $\square$

BGP policy routing is, naturally, substantially richer than $\mathcal{B}_1$ or $\mathcal{B}_2$. At the third level, for instance, usually path length is taken into account, leading to the algebra $\mathcal{B}_3 = \mathcal{B}_2 \times \mathcal{S}$. Using the foregoing argumentation, one easily checks that $\mathcal{B}_3$ is also incompressible.

## 6 Conclusions and open questions

Thanks to the tenacious research efforts in the field of compact routing, we now have a remarkable insight into the theoretical scalability of shortest path routing. Motivated by the fact that many routing applications adopt a significantly more complex way to classify paths than pure shortest path routing (for instance, BGP places path length only at the third place when fixing path preference), in this paper we proposed an algebraic approach towards generalizing the theory of compact routing to policy routing. Our contribution is twofold: first, we presented some "landmark" theorems, which can be used as guidelines to roughly classify routing policies based on their algebraic properties, and second we identified some algebraic requirements for effectively trading between path preference and memory. As an important message, we identified regularity as the cornerstone of compact policy routing, allowing for a generic compressibility theory to be formulated as well as defining a finite stretch compact routing scheme. The fact that regular algebras are exactly the ones that can be efficiently implemented in a distributed way [19–22] makes these algebras highly attractive for designing future routing policies [31].

Besides answering the most elemental questions, this paper perhaps leaves more issues open than it answers. We have seen that selectivity is sufficient for a routing algebra to be compressible, and strict monotonicity is sufficient for incompressibility. However, it is not clear which are the corresponding necessary conditions. Easily, strict monotonicity is not necessary for incompressibility as evidenced by the non-monotone $\mathcal{B}_2$ algebra. Finding a minimal algebra that eventuates incompressibility is therefore an interesting open issue. On the other hand, by requiring selectivity for compressibility we seem to be on the safe side, since selectivity not only guarantees compressibility but also a very appealing memory requirement of $O(\log n)$. Whether there are compressible algebras with $\Omega(\log n)$ local memory requirement is also an intriguing problem. As pointed out in the paper, it is also an open question whether the $\Omega(n)$ characterization for non-isotone algebras is tight, as the only trivial routing function needs $O(n^2 \log d)$ bits per router.

We have shown some real-world routing policies whose memory requirement cannot be relaxed, even by allowing arbitrary finite stretch. Unfortunately, the widely applied BGP policy qualifies for this property. Therefore, perhaps the most compelling question raised in this paper is "what can we do if stretch doesn't help?"

## Acknowledgements

This work was performed in the High Speed Networks Laboratory at BME-TMIT. This work is connected to the scientific program of the "Development of quality-oriented and cooperative R+D+I strategy and functional model at BME" project. This project is supported by the New Hungary Development Plan (Project ID: TÁMOP-4.2.1/B-09/1/KMR-2010-0002).

## References

[1] P. Fraigniaud and C. Gavoille. Memory requirement for universal routing schemes. In *Proceedings of the fourteenth annual ACM symposium on Principles of distributed computing*, PODC '95, pages 223–230, 1995.

[2] C. Gavoille and S. Pérennès. Memory requirement for routing in distributed networks. In *Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing*, PODC '96, pages 125–133, 1996.

[3] L. Cowen. Compact routing with minimum stretch. In *ACM-SIAM SODA'99*, pages 255–260, 1999.

[4] M. Thorup and U. Zwick. Compact routing schemes. In *ACM SPAA'01*, pages 1–10, 2001.

[5] C. Gavoille. Routing in distributed networks: Overview and open problems. *ACM SIGACT News*, 32(1):52, 2001.

[6] D. Krioukov, kc claffy, K. Fall, and A. Brady. On compact routing for the Internet. *ACM Comp. Comm. Review*, 37(3):41–52, 2007.

[7] C. Gavoille. An overview on compact routing. In *Workshop on Peer-to-Peer, Routing in Complex Graphs, and Network Coding*, 2007.

[8] G.N. Frederickson and R. Janardan. Designing networks with compact routing tables. *Algorithmica*, 3(1):171–190, 1988.

[9] D. Krioukov, K. Fall, and X. Yang. Compact routing on Internet-like graphs. In *INFOCOM 2004, the Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 1, 2004.

[10] P. Fraigniaud and C. Gavoille. Routing in trees. In *ICALP '01*, pages 757–772, 2001.

[11] O. Younis and S. Fahmy. Constraint-based routing in the Internet: Basic principles and recent research. *IEEE Communications Surveys and Tutorials*, 5(1), 2004.

[12] Zheng Wang and Jon Crowcroft. Quality-of-service routing for supporting multimedia applications. *IEEE Journal of Selected Areas in Communications*, 14(7):1228–1234, 1996.

[13] G. Apostolopoulos, R. Guerin, S. Kamat, and S. K. Tripathi. Quality of service based routing: A performance perspective. In *SIGCOMM*, pages 17–28, 1998.

[14] Qingming Ma and P. Steenkiste. On path selection for traffic with bandwidth guarantees. In *Proceedings of the 1997 International Conference on Network Protocols (ICNP '97)*, page 191, 1997.

[15] M. Caesar and J. Rexford. BGP routing policies in ISP networks. Technical Report UCB/CSD-05-1377, EECS Department, University of California, Berkeley, 2005.

[16] G. Apostolopoulos, R. Guerin, S. Kamat, A. Orda, and S. K. Tripathi. Intra-domain QoS routing in IP networks: A feasibility and cost/benefit analysis. *IEEE Network*, 13:42–54, 1999.

[17] D. Awduche. MPLS and traffic engineering in IP networks. *IEEE Communications Magazine*, 37(12):42–47, Dec 1999.

[18] W. Lee, M. Hluchyi, and P. Humblet. Routing subject to quality of service constraints in integrated communication networks. *IEEE Network Magazine*, 9(4):46–55, July-August 1999.

[19] J. Sobrinho. Algebra and algorithms for QoS path computation and hop-by-hop routing in the Internet. *IEEE/ACM Trans. Netw.*, 10:541–550, August 2002.

[20] J. Sobrinho. Network routing with path vector protocols: theory and applications. In *SIGCOMM '03*, pages 49–60, 2003.

[21] T. Griffin and J. Sobrinho. Metarouting. In *SIGCOMM '05*, pages 1–12, 2005.

[22] A. Gurney and T. Griffin. Lexicographic products in metarouting. In *Network Protocols, IEEE International Conference on*, pages 113–122, 2007.

[23] C.-K. Chau, R. Gibbens, and T. G. Griffin. Towards a unified theory of policy-based routing. In *INFOCOM 2006, the 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–12, 2006.

[24] M. Gondran and M. Minoux. *Graphs, Dioids and Semirings: New Models and Algorithms.* Springer Publishing Company, Incorporated, 1 edition, 2008.

[25] B. Awerbuch and Y. Shavitt. Topology aggregation for directed graphs. *IEEE/ACM Trans. Netw.*, 9:82–90, February 2001.

[26] A. H. Clifford and G. B. Preston. *The Algebraic Theory of Semigroups, Volume I.* Number 7 in Mathematical Surveys. American Mathematical Society, 1961.

[27] G. Huston. Interconnection, peering, and settlements. In *Proceedings of the INET*, 1999.

[28] F. Wang and L. Gao. On inferring and characterizing Internet routing policies. In *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 15–26, 2003.

[29] L. Gao. On inferring autonomous system relationships in the Internet. *IEEE/ACM Trans. on Networking*, 9:733–745, 2000.

[30] Pierre Fraigniaud and Cyril Gavoille. Local memory requirement of universal routing schemes. Technical Report 96-01, École Normale Supérieure de Lyon, 69364 Lyon Cedex 07, 1996.

[31] A. Seehra, J. Naous, M. Walfish, D. Mazieres, A. Nicolosi, and S. Shenker. A policy framework for the future Internet. *HotNets-VIII*, 2009.