Routing on Multiple Optimality Criteria

João Luís Sobrinho
Instituto de Telecomunicações
Instituto Superior Técnico, Universidade de Lisboa

Miguel Alves Ferreira
Instituto de Telecomunicações
Instituto Superior Técnico, Universidade de Lisboa

ABSTRACT

Standard vectoring protocols, such as EIGRP, BGP, DSDV, or Babel, only route on optimal paths when the total order on path attributes that substantiates optimality is consistent with the extension operation that calculates path attributes from link attributes, leaving out many optimality criteria of practical interest. We present a solution to this problem and, more generally, to the problem of routing on multiple optimality criteria. A key idea is the derivation of a partial order on path attributes that is consistent with the extension operation and respects every optimality criterion of a designated collection of such criteria. We design new vectoring protocols that compute on partial orders, with every node capable of electing multiple attributes per destination rather than a single attribute as in standard vectoring protocols. Our evaluation over publicly available network topologies and attributes shows that the proposed protocols converge fast and enable optimal path routing concurrently for many optimality criteria with only a few elected attributes at each node per destination. We further show how predicating computations on partial orders allows incorporation of service chain constraints on optimal path routing.

CCS CONCEPTS

• **Networks** → **Routing protocols**; *Network simulations.*

KEYWORDS

Routing, optimal path routing, optimality criteria, routing algebras, partial orders, routing protocols.

ACM Reference Format:
João Luís Sobrinho and Miguel Alves Ferreira. 2020. Routing on Multiple Optimality Criteria. In *Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication (SIGCOMM '20), August 10–14, 2020, Virtual Event, NY, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3387514.3405864

1 INTRODUCTION

The concept of optimal path is bound to: (1) a set of attributes, which represents performance metrics in context; and (2) a total order on attributes, which defines relative preferences among them. The optimal attribute from a source to a destination in a network is the most preferred of all path attributes from source to destination and an optimal path is one with such an attribute. A binary extension operation on attributes allows the calculation of the attribute of a path from the attributes of its constituent links [7, 15, 18, 21, 35, 36, 43, 45].

Standard vectoring protocols, such as EIGRP [34], BGP [33], DSDV [30], or Babel [11], iterate at every node of a network: (1) extension operations, which compute attributes to reach destinations from attributes advertised by neighbors; and (2) selection operations in accordance with the total order, which elect a single attribute per destination. This approach only discovers optimal attributes and paths if the total order is consistent with the extension operation or, in more precise terms, if the extension operation is isotone for the total order, meaning that the relative preference between any two attributes is preserved when both are extended by any third attribute [7, 15, 35]. Without isotonicity, standard vectoring protocols fail to route on optimal paths, in general [16, 35, 37].

However, for the most part, total orders representing real-world performance metrics do not satisfy isotonicity. For example, a quickest path is desired to convey a file across a network, which is a path that minimizes a linear combination of propagation delay and inverse capacity [9]. Yet, the total order underlying quickest paths is not isotone [16]. For another example, the choice of a path on which to stream a video across a network weighs minimal delay against sufficient available bandwidth to sustain the stream [2]. Yet again, the total order framing such a choice is not isotone.

1.1 Contribution

The main contribution of this paper is a general solution to the problem of optimal path routing concurrently for multiple of optimality criteria, which entails a general solution to the problem of optimal path routing for a single non-isotone optimality criterion.

Optimal path routing for a non-isotone criterion. The solution to this problem is based on two novel ideas. The first is the substitution of a total order on attributes by a partial order that satisfies isotonicity while respecting the total order. In a partial order, one attribute of a pair of attributes is preferred to the other or the two attributes are incomparable [20]. The set of dominant attributes from a source to a destination in a network consists of those path attributes with no path attribute from source to destination preferred to any of them. It is a plural set of pairwise incomparable attributes, in general, which contains the original optimal attribute.

The second idea is the design of dominant-paths vectoring protocols that compute on partial orders. These protocols instantiate a separate computation process per destination, as standard vectoring protocols do, but have every node elect and advertise to neighbors a set of dominant attributes to reach the destination rather than a single most preferred attribute. Every node labels each attribute of its elected set with a locally unique identifier that is advertised alongside the attribute [8]. A source of data-packets recognizes the original optimal attribute among its elected set of dominant

SIGCOMM '20, August 10–14, 2020, Virtual Event, NY, USA

ACM ISBN 978-1-4503-7955-7/20/08…$15.00
https://doi.org/10.1145/3387514.3405864

attributes and data-packets are forwarded along a corresponding path through label-switching at intermediate nodes.

Optimal path routing for multiple criteria. The solution to this problem is further based on the idea of intersecting the total orders of a designated collection of optimality criteria to produce a partial order that satisfies isotonicity while respecting all total orders of the collection. Then, a dominant-paths vectoring protocol provides optimal path routing concurrently for all criteria. For every flow of data-packets, the source chooses the optimality criterion most appropriate to route that specific flow on to the destination, with the corresponding optimal attribute found among the set of dominant attributes computed by the protocol.

The routing state maintained by a dominant-paths vectoring protocol on a network is proportional to the sizes of the sets of dominant attributes from sources to destinations. Our solution to routing on multiple optimality criteria is practical to the extent that these sets are small. We computed sets of dominant attributes on annotated Rocketfuel topologies [39]. The results show that the average number of dominant attributes from source to destination is below four even for a network with hundreds of nodes and more than a thousand links.

Another important consideration is the speed of convergence of a dominant-paths vectoring protocol. The multiple attributes comprising a set of dominant attributes are elected in parallel during an execution of the protocol. Furthermore, we found that isotonicity promotes fast convergence. Our simulations on the Rocketfuel topologies confirm that the convergence time of a dominant-paths vectoring protocol operating on an isotone partial order is only marginally worse than that of a standard vectoring protocol operating on an isotone total order, and that it is sometimes better than that of a standard vectoring protocol operating on a non-isotone total order.

1.2 Roadmap

Routing on a non-isotone optimality criterion and on multiple optimality criteria is first illustrated in Section 2. Section 3 develops a procedure that starts with a generic collection of optimality criteria and ends with a partial order that respects each criterion and satisfies isotonicity. Then, Section 4 designs two classes of vectoring protocols that compute on partial orders. Section 5 shows how these protocols can accommodate service chaining constraints. An evaluation of our solution to routing on multiple optimality criteria is presented in Section 6. Section 7 reviews related work and Section 8 concludes the paper. The appendices contain proofs of termination and dominance for vectoring protocols.

This work does not raise any ethical issues.

2 ROUTING ON WIDTHS AND LENGTHS

We illustrate how vectoring protocols based on partial orders allow routing on a variety of optimality criteria. In the forthcoming examples, every link and path in a network is characterized by a pair *width-length* belonging to the *Cartesian product* of positive or infinite widths and nonnegative lengths. Width represents a metric, such as capacity or available bandwidth, that extends along a path with the minimum operator, whereas length represents a metric, such as delay or number of data-packets in queue, that extends along a path with addition. Therefore, the *extension* of width-length (w, l) with width-length (w', l') is width-length $(\min\{w, w'\}, l + l')$.

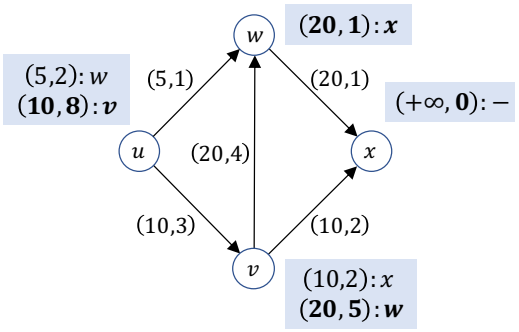


Figure 1: Stable state of a standard vectoring protocol operating according to the shortest-widest order for destination x. Links are annotated with width-lengths. Elected width-lengths are in bold.

Section 2.1 discusses shortest-widest path routing and Section 2.2 continues the discussion with widest-shortest path routing.

2.1 Shortest-widest path routing

A *shortest-widest path* is a path of minimum length among those of maximum width from source to destination in a network [43].[1] Shortest-widest paths are selected according to the *shortest-widest order* (lexicographic order), which establishes that width-length (w, l) is preferred to width-length (w', l') if its width is greater, $w > w'$, or the widths are equal but its length is smaller, $w = w'$ and $l < l'$.

In the network of Figure 1, each link is annotated with a pair width-length and all nodes want to route data-packets to destination x along shortest-widest paths. By inspection, we readily conclude that the shortest-widest path from v to x is vwx, with width-length $(20, 5) = (\min\{20, 20\}, 4+1)$, and that the shortest-widest path from u to x is path uvx, with width-length $(10, 5) = (\min\{10, 10\}, 3+2)$.

With a standard vectoring protocol, each node elects and advertises to its in-neighbors the most preferred width-length learned from its out-neighbors. The stable state of such a protocol is shown in the figure. Destination x elects $(+\infty, 0)$ and w elects $(20, 1)$. Node v learns $(10, 2)$ from x and $(20, 5)$ from w, which is the extension of $(20, 4)$ of link vw with $(20, 1)$ of the elected width-length at w. It elects $(20, 5)$, learned from w, instead of $(10, 2)$, learned from x, on account of its greater width. Node u learns $(5, 2)$ from w, which is the extension of $(5, 1)$ with $(20, 1)$, and $(10, 8)$ from v, which is the extension of $(10, 3)$ with $(20, 5)$. It elects $(10, 8)$, learned from v, because of its greater width.

Node u forwards data-packets to v, which forwards them to w, which delivers them to x. Thus, data-packets with source at u and destination at x travel along path $uvwx$, which is not the shortest-widest path from u to x. The standard vectoring protocol fails to route data-packets along shortest-widest paths. This is due to the failure of isotonicity of extension for the relative preferences among width-lengths [35]. Concretely, $(20, 5)$ is preferred to $(10, 2)$, but

[1] This is the standard definition, even if reading from left to right may wrongly suggest that shortest paths are selected first.

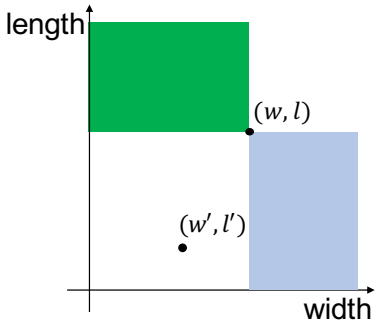


Figure 2: Product order on width-lengths. The green area consists of width-lengths that are less preferred than (w, l) and the blue area of those that are preferred to (w, l). Width-lengths (w, l) and (w', l') are incomparable on the product order, but (w, l) is preferred to (w', l') on the shortest-widest order.

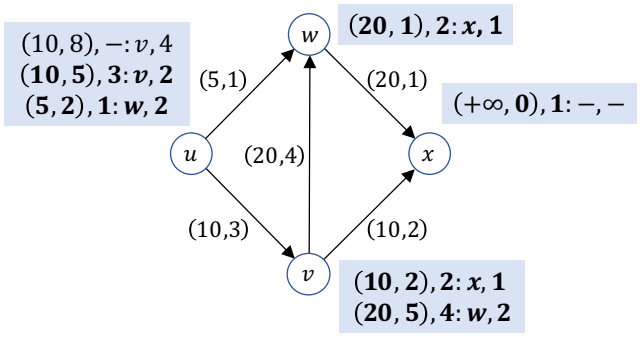


Figure 3: Stable state of a dominant-paths vectoring protocol operating according to the product order on width-lengths for destination x. Links are annotated with width-lengths. Labels guide data-packets along shortest-widest paths or widest-shortest paths as decided by the source.

$(10, 8)$, which is the extension of $(10, 3)$ with $(20, 5)$, is less preferred than $(10, 5)$, which the extension of $(10, 3)$ with $(10, 2)$.

Now consider the *product order* on width-lengths [20], which is such that width-length (w, l) is preferred to width-length (w', l') if it is different from (w', l') and both its width equals or is greater, $w \geq w'$, and its length equals or is smaller, $l \leq l'$, than those of (w', l'). The product order is a partial order. Two width-length such that one has greater width but the other has smaller length are *incomparable*, neither of them being preferred to the other. Figure 2 shows the width-length plane where the set of width-lengths that are less preferred than (w, l) on the product order is shaded in green and the set of width-lengths that are preferred to (w, l) is shaded in blue. Width-lengths (w, l) and (w', l') are incomparable, yet (w, l) is preferred to (w', l') on the shortest-widest order on account of its greater width.

A width-length in a set of width-lengths is *dominant* if no width-length in the set is preferred to it.[2] A *dominant path* is one whose width-length is dominant among the width-lengths of all paths from a source to a destination in a network. Figure 3 shows the same network as Figure 1. The dominant paths from v to x are vx and vwx. Their width-lengths, respectively, $(10, 2)$ and $(20, 5)$, are incomparable. The dominant paths from u to x are uvx and uwx, width-lengths $(10, 5)$ and $(5, 2)$, respectively. Path $uvwx$, the only remaining path from u to x, has width-length $(10, 8)$, which is less preferred than width-length $(10, 5)$ of path uvx.

A *dominant-paths vectoring protocol* computes on the product order, each node electing and advertising to in-neighbors the set of dominant width-lengths learned from out-neighbors. The stable state of such a protocol is shown in Figure 3. Node x elects $(+\infty, 0)$ and w elects $(20, 1)$ as before. Node v learns $(10, 2)$ from x and $(20, 5)$ from w. Since these width-lengths are incomparable, both are elected. Node v differentiates the two width-lengths by assigning them distinct labels. It assigns label 2 to $(10, 2)$ and label 4 to $(20, 5)$. These labels are advertised alongside the associated width-lengths to in-neighbor u in order to enable expedition of data-packets via label-switching. Therefore, u learns from v both $(10, 5)$ with label 2 and $(10, 8)$ with label 4. Both become candidate width-lengths to reach x via v. From w, u also learns $(5, 2)$ with label 2. The dominant width-lengths of the set $\{(10, 5), (10, 8), (5, 2)\}$ of candidate width-lengths at u are $(10, 5)$ and $(5, 2)$, since these two width-lengths are incomparable, while $(10, 8)$ is less preferred than $(10, 5)$. Node u elects $(10, 5)$ and $(5, 2)$, assigning label 3 to the former width-length and label 1 to the latter.

In the figure, an entry at a node of the form

$$(width, length)\,,\ label\ :\ next.hop\,,\ next.label$$

reads as follows:

- data-packets sourced at the node that need to travel along a path with width-length $(width, length)$ are forwarded to out-neighbor $next.hop$ with label $next.label$;
- data-packets arriving at the node from an in-neighbor carrying label $label$ are forwarded to out-neighbor $next.hop$ with the label modified to $next.label$.

With the dominant-paths vectoring protocol, u has routing and forwarding information to see that its data-packets traverse a shortest-widest path to x. Width-length $(10, 5)$ is the shortest-widest width-length elected at u. Thus, u stamps data-packets with label 2 and forwards them to v. At v, incoming label 2 matches the entry pointing to out-neighbor x and outgoing label 1. Consequently, v replaces label 2 with label 1 and forwards the data-packets to x.

2.2 Widest-shortest path routing

A *widest-shortest path* is a path of maximum width among those of minimum length from source to destination in a network [43]. Widest-shortest paths are selected according to the *widest-shortest order* (colexicographic order), which establishes that width-length (w, l) is preferred to width-length (w', l') if its length is smaller, $l < l'$, or the lengths are equal and its width is greater, $l = l'$ and $w > w'$.

The dominant-paths vectoring protocol allows for routing a flow on a shortest-widest path or on a widest-shortest path as is more

[2] In the terminology of order theory, a dominant width-length is a *minimal* width-length with respect to the product order.

appropriate for that specific flow. Going back to Figure 3, suppose that u now wants to send data-packets to x along a widest-shortest path. Width-length $(5, 2)$ is the widest-shortest width-length at u to reach x. Reading from the entry corresponding to $(5, 2)$, u stamps data-packets with label 2 and forwards them to w, which replaces label 2 with label 1 for delivery to x.

Forthcoming Sections 3 and 4 present the concepts, constructs, and protocols that allow routing on multiple optimality criteria for arbitrary performance metrics.

3 MULTIPLE OPTIMALITY AND DOMINANCE

We formulate the problem of routing on multiple optimality criteria with generality and develop constructs that facilitate a solution to this problem. Section 3.1 introduces basic algebraic routing concepts and presents an algebraic statement for the problem of routing on multiple optimality criteria. Section 3.2 exemplifies with two families of optimality criteria on pairs width-length. Section 3.3 introduces partial orders and the possibility of reducing them so that the algebraic property of isotonicity is satisfied. Section 3.4 transforms the problem of routing on multiple optimality criteria into that of routing based on a partial order that respects all criteria and satisfies isotonicity. Section 3.5 applies the constructions expounded in the previous two sections to examples of optimality criteria on pairs width-length.

3.1 Optimality criteria

Optimal path problems can be formulated with generality in algebraic terms [7, 15, 18, 35]. A set S of *attributes* represents arbitrary performance metrics. To every link and path in a network is associated an attribute. The attribute of a path is obtained from the attributes of its constituent links through a *binary extension operation*, denoted by $\oplus$, that we assume to be associative and commutative with neutral attribute ϵ. Letting $a[uv]$ denote the attribute of link uv, the attribute $a[P]$ of path $P = u_0u_1 \cdots u_{n-1}u_n$ is given by

$$a[P] = a[u_0u_1] \oplus \cdots \oplus a[u_{n-1}u_n].$$

The attribute of a trivial path, containing just one node, is ϵ.

We consider a collection O of optimality criteria. Optimality criterion $i \in O$ is modeled by a *total order* $\preceq_i$ on attributes, which is an antisymmetric, transitive, and connex binary relation on the set of attributes. Connexity means that $a \preceq_i b$ or $b \preceq_i a$ for all $a, b \in S$. We write $a \prec_i b$ for $a \preceq_i b$ and $a \neq b$, and say that a is *i-preferred* to b and that b is *less i-preferred* than a. The null attribute $\bullet$ is the least preferred of all attributes and represents the absence of a valid path.

Given a network, the *i-optimal* attribute from source s to destination t, denoted by $a_i^*(s, t)$, is the most i-preferred attribute among all path attributes from s to t. A path from s to t is i-optimal if its attribute is $a_i^*(s, t)$.

Definition 3.1. Binary extension operation $\oplus$ is *inflationary* for total order $\preceq_i$ if

$$b \preceq_i a \oplus b \text{ for all } a, b \in S.$$

Inflation expresses that the attribute of a path is not i-preferred to the attribute of any of its subpaths [17, 35].[3] It is typically satisfied by performance metrics and, therefore, it is assumed throughout the paper. Inflation is related to the termination (convergence) of standard vectoring protocols in stable states that guide data-packets from sources to destinations, though not necessarily along i-optimal paths. The exact nature of this relationship depends on the class of vectoring protocols considered and is discussed further in Section 4.

Definition 3.2. Binary extension operation $\oplus$ is *isotone* for total order $\preceq_i$ if

$$a \preceq_i b \text{ implies } c \oplus a \preceq_i c \oplus b \text{ for all } a, b, c \in S.$$

Isotonicity expresses that the relative i-preference between the attributes of two paths is preserved when both are prefixed by any common third path [7, 16, 35]. Contrary to inflation, isotonicity is only satisfied by a very restricted set of performance metrics (see next section). When isotonicity holds, a standard vectoring protocol routes data-packets on i-optimal paths; when it does not hold, data-packets are not routed on i-optimal paths, in general [16, 35].

3.2 Examples

We present two parameterized families of optimality criteria based on width-lengths.

K-quickest order. The time required to convey a file of size K along a path with capacity w and propagation delay l is $K/w + l$. A *K-quickest path* from source to destination in a network is one that minimizes the time required to convey such a file [9]. Accordingly, the *K-quickest order*, $\preceq_{K\text{-Q}}$, is defined by

$$(w, l) \preceq_{K\text{-Q}} (w', l') \text{ if:}$$
$$K/w + l < K/w' + l', \text{ or } K/w + l = K/w' + l' \text{ and } w \geq w'.$$

W-wide-shortest order. In order to stream a video of maximum rate W from source to destination, a network operator would typically choose a *W-wide-shortest path*, which is a path of minimum delay among those with available bandwidth higher than or equal to W; if the bandwidth available on every path is lower than W, then it is a path of maximum available bandwidth [2]. Motivated by this scenario, we define the *W-wide-shortest order*, $\preceq_{W\text{-S}}$, as a total order that corresponds to the widest-shortest order for widths higher than or equal to W and to the shortest-widest order for widths lower than W:

$$\begin{aligned}(w, l) \preceq_{W\text{-S}} (w', l') \text{ if:}& \\ w \geq W \text{ and } (w' < W \text{ or } (w, l) \preceq_{\text{WS}} (w', l'))&; \\ \text{or } w < W \text{ and } (w, l) \preceq_{\text{SW}} (w', l')&,\end{aligned}$$

where $\preceq_{\text{WS}}$ and $\preceq_{\text{SW}}$ denote, respectively, the widest-shortest and the shortest-widest orders.

Inflation of the criteria. Both the K-quickest order and the W-wide-shortest order satisfy inflation.

Isotonicity of the criteria. Neither the K-quickest order nor the W-wide-shortest order satisfy isotonicity, except for the limiting cases of $K = 0$ and $W = 0$ (widest-shortest order). Routing optimally on these criteria is not possible with standard vectoring protocols.

3.3 Partial orders and isotonic reductions

A *partial order* $\preceq$ on attributes is an antisymmetric, transitive, and reflexive binary relation on attributes. Reflexivity means that $a \preceq a$ for all $a \in S$. Connexity implies reflexivity, so that a total order is a particular case of a partial order. If $a \preceq b$ or $b \preceq a$, then a and b are

[3] Some authors use the term *monotonicity* for what we call inflation.

comparable; otherwise they are *incomparable*. We still write $a \prec b$ for $a \preceq b$ and $a \neq b$, and say that a is *preferred* to b and that b is *less preferred* than a.

Given a network, the set of *dominant* attributes from source s to destination t, denoted by $A^*(s,t)$, is the set of path attributes from s to t such that no path attribute from s to t is preferred to any of the attributes in the set. The attributes of $A^*(s,t)$ are pairwise incomparable. A path from s to t is *dominant* if its attribute belongs to $A^*(s,t)$.

A key idea for routing on dominant paths through the use of vectoring protocols is to identify a partial order that satisfies isotonicty within a larger partial order that does not satisfy it. This idea is embodied in the following definition [25].

Definition 3.3. An *isotonic reduction* of a partial order $\preceq$ on attributes is a partial order contained in $\preceq$ for which $\oplus$ is isotone.

The more attributes that can be compared, the more efficient the vectoring protocols that we present in Section 4. Hence, we aspire to isotonic reductions with as many comparable pairs of attributes as possible.

Theorem 3.4. *Every partial order on attributes $\preceq$ contains a largest isotonic reduction $\preceq_R$, which is such that for every pair of attributes a and b, $a \preceq_R b$ if and only if $x \oplus a \preceq x \oplus b$ for every attribute x.*

Proof. We show that the binary relation on attributes $\preceq_R$ defined by $a \preceq_R b$ if $x \oplus a \preceq x \oplus b$ for all $x \in S$ is the largest isotonic reduction of $\preceq$ on S.

Binary relation $\preceq_R$ is a partial order. (1) Reflexivity: $a \preceq_R a$, because $x \oplus a \preceq x \oplus a$ for all $x \in S$. (2) Antisymmetry: $a \preceq_R b$ and $b \preceq_R a$ imply $a = b$, because $a \preceq_R b$ implies $a = \epsilon \oplus a \preceq \epsilon \oplus b = b$, $b \preceq_R a$ likewise implies $b \preceq a$, and $a \preceq b$ together with $b \preceq a$ imply $a = b$. (3) Transitivity: $a \preceq_R b$ and $b \preceq_R c$ imply $a \preceq_R c$, because $x \oplus a \preceq x \oplus b$ and $x \oplus b \preceq x \oplus c$ together imply $x \oplus a \preceq x \oplus c$ for all $x \in S$, which implies $a \preceq_R c$.

Binary extension operation $\oplus$ is isotone for $\preceq_R$. The inequality $a \preceq_R b$ implies $(x \oplus c) \oplus a \preceq (x \oplus c) \oplus b$ for all $x, c \in S$. Using associativity of $\oplus$, we write $(x \oplus c) \oplus a = x \oplus (c \oplus a)$ and $(x \oplus c) \oplus b = x \oplus (c \oplus b)$, so that $x \oplus (c \oplus a) \preceq x \oplus (c \oplus b)$ for all $x, c \in S$, which implies $c \oplus a \preceq_R c \oplus b$ for all $c \in S$.

Partial order $\preceq_R$ is the largest isotonic reduction of $\preceq$ on attributes. In order to arrive at a contradiction, suppose that there is a partial order $\preceq'_R$ contained in $\preceq$, but not strictly contained in $\preceq_R$. Therefore, there is $a \preceq'_R b$ such that $a \preceq b$, while it is not the case that $a \preceq_R b$. Thus, there is an attribute x for which it is not the case that $x \oplus a \preceq x \oplus b$. If $\oplus$ were isotone for $\preceq'_R$, then we would have $x \oplus a \preceq'_R x \oplus b$, which would imply $x \oplus a \preceq x \oplus b$: a contradiction was arrived at. □

Inflation remains a desirable property, which is related to the termination of the vectoring protocols that we present in Section 4 into stable states that guide data-packets from sources to destinations. We have the following theorem.

Theorem 3.5. *The largest isotonic reduction of an inflationary partial order is itself inflationary.*

Proof. Let $\preceq$ be a partial order on S for which $\oplus$ is inflationary and denote by $\preceq_R$ the largest isotonic reduction of $\preceq$ on S. In order to show that $b \preceq_R a \oplus b$, we need to assert that $x \oplus b \preceq x \oplus (a \oplus b)$ for all $x \in S$. Because $\oplus$ is inflationary for $\preceq$, we have $x \oplus b \preceq a \oplus (x \oplus b)$ for all $x \in S$. Using associativity and commutativity of $\oplus$, we write $a \oplus (x \oplus b) = x \oplus (a \oplus b)$, so that $x \oplus b \preceq x \oplus (a \oplus b)$ for all $x \in S$. □

3.4 From optimality to dominance

We present a procedure that, from the collection O of optimality criteria described in Section 3.1, yields a partial order on attributes that respects all criteria and satisfies isotonicity.

First, we define binary relation on attributes $\preceq_O$ as the *intersection* of total orders $\preceq_i$, $i \in O$. For every pair of attributes a and b,

$$a \preceq_O b \text{ if } a \preceq_i b \text{ for all } i \in O.$$

It is easy to verify that binary relation $\preceq_O$ is a partial order. Two attributes are comparable in $\preceq_O$ if and only if one of them is i-preferred to the other for all criteria i.

Second, if $\oplus$ is not isotone for $\preceq_O$, then we retain only the largest isotonic reduction of $\preceq_O$, as prescribed by Theorem 3.4, which we denote by $\preceq_{O,R}$. Since $\oplus$ is inflationary for $\preceq_i$, $i \in O$, $\oplus$ is inflationary for $\preceq_O$. Then, from Theorem 3.5, we conclude that $\oplus$ is inflationary for $\preceq_{O,R}$.

Given a network, let $A^*_{O,R}(s,t)$ denote the set of dominant attributes from source s to destination t according to partial order $\preceq_{O,R}$. By construction, $\preceq_{O,R}$ is contained in every total order $\preceq_i$, $i \in O$. In other words, $a \preceq_{O,R} b$ implies $a \preceq_i b$ for all $a, b \in S$ and all $i \in O$. Therefore, the i-optimal attribute from s to t is one of the dominant attributes from s to t:

$$a^*_i(s,t) \in A^*_{O,R}(s,t) \text{ for all } i \in O.$$

In summary, the problem of computing i-optimal attributes, $a^*_i(s,t)$, for all criteria i has been transformed into the problem of computing sets of dominant attributes determined according to the largest isotonic reduction of the intersection of all criteria, $A^*_{O,R}(s,t)$.

3.5 Examples revisited

We apply the constructions of the previous two sections to obtain largest isotonic reductions and intersections for the total orders on width-lengths that we have been using as examples.

Proposition 1. *The largest isotonic reduction of the shortest-widest order is the product order on width-lengths.*

Proof. We make use of the characterization of largest isotonic reduction provided by Theorem 3.4. Denote the product order on width-lengths by $\preceq_{\mathrm{W}\times\mathrm{L}}$. Clearly, $(w,l) \preceq_{\mathrm{W}\times\mathrm{L}} (w',l')$ implies $(\min(x,w), m+l) \preceq_{\mathrm{SW}} (\min(x,w'), m+l')$ for every width-length (x,m).

Conversely, suppose that $(w,l) \preceq_{\mathrm{W}\times\mathrm{L}} (w',l')$ does not hold. Hence, $w < w'$ or $l > l'$. We need to show that there is width-length (x,m) such that $(\min(x,w), m+l) \preceq_{\mathrm{SW}} (\min(x,w'), m+l')$ does not hold. If $w < w'$, then it is not the case that $(w,l) \preceq_{\mathrm{SW}} (w',l')$. We choose $(x,m) = (+\infty, 0)$ to conclude that

$$(\min(+\infty,w), 0+l) = (w,l) \preceq_{\mathrm{SW}} (w',l') = (\min(+\infty,w'), 0+l')$$

does not hold. Otherwise, if $w \geq w'$, then it must be the case that $l > l'$. We choose $(x,m) = (w',1)$ to conclude that

$$(\min(w',w), 1+l) = (w', 1+l) \preceq_{\mathrm{SW}} (w', 1+l') = (\min(w',w'), 1+l')$$

does not hold. □

Routing optimally on shortest-widest paths is possible by instantiating the dominant-paths vectoring protocols presented in Section 4 with the product order on width-lengths.

Proposition 2. *The largest isotonic reduction of the K-quickest order equals the intersection of the K-quickest order and the widest-shortest order. Specifically, width-length (w, l) equals or is preferred to width-length (w', l') in the largest isotonic reduction of $\preceq_{K\text{-Q}}$ if and only if $(w, l) \preceq_{K\text{-Q}} (w', l')$ and $l \leq l'$.*

Proof. We again make use of Theorem 3.4. First, we show that $(w, l) \preceq_{K\text{-Q}} (w', l')$ and $l \leq l'$ together imply $(\min(x, w), m + l) \preceq_{K\text{-Q}} (\min(x, w'), m + l')$ for every width-length (x, m). Two cases are distinguished.

Case 1: $w \geq w'$. We have $\min(x, w) \geq \min(x, w')$. From $l \leq l'$, we write

$$K/\min(x, w) + m + l \leq K/\min(x, w') + m + l'.$$

Consequently, $(\min(x, w), m + l) \preceq_{K\text{-Q}} (\min(x, w'), m + l')$.

Case 2: $w < w'$. From $(w, l) \preceq_{K\text{-Q}} (w', l')$ and $w < w'$, we deduce that $l < l'$. If $x \leq w$, then $x = \min(x, w) = \min(x, w')$, and we write

$$K/\min(x, w) + m + l < K/\min(x, w') + m + l',$$

so that $(\min(x, w), m + l) \preceq_{K\text{-Q}} (\min(x, w'), m + l')$. If $w < x < w'$, then $w = \min(x, w) < x = \min(x, w')$. From $(w, l) \preceq_{K\text{-Q}} (w', l')$, we write

$$\begin{aligned} K/\min(x, w) + m + l &= K/w + m + l \\ &\leq K/w' + m + l' \\ &< K/\min(x, w') + m + l'. \end{aligned}$$

Once again, $(\min(x, w), m + l) \preceq_{K\text{-Q}} (\min(x, w'), m + l')$. Last, if $w' \leq x$, then widths w and w' are not diminished by width x. We obtain $(\min(x, w), m + l,) \preceq_{K\text{-Q}} (\min(x, w'), m + l')$ directly from $(w, l) \preceq_{K\text{-Q}} (w', l')$.

Second, we show that if either $(w, l) \preceq_{K\text{-Q}} (w', l')$ does not hold or $l > l'$, then there is width-length (x, m) such that $(\min(x, w), m + l) \preceq_{K\text{-Q}} (\min(x, w'), m + l')$ does not hold. If $(w, l) \preceq_{K\text{-Q}} (w', l')$ does not hold, then we simply choose $(x, m) = (+\infty, 0)$. Otherwise, if $l > l'$, then we choose $(x, m) = (\min(w, w'), 1)$ to obtain $K/\min(w, w') + 1 + l > K/\min(w, w') + 1 + l'$, which implies that $(\min(x, w), m + l) \preceq_{K\text{-Q}} (\min(x, w'), m + l')$ does not hold. □

The largest isotonic reduction of the K-quickest order is larger than the product order on width-lengths and it grows as the value of K gets smaller. In the limiting case of $K = 0$ (widest-shortest order), all pairs of width-lengths are comparable. Larger orders mean more comparable pairs of width-lengths, fewer dominant width-lengths from source to destination in a network, and more efficient vectoring protocols.

The following two easy propositions are presented without proof.

Proposition 3. *The intersection of the K-quickest orders for all $K \geq 0$ is the product order on width-lengths.*

Proposition 4. *The intersection of the W-wide-shortest orders for all $W \geq 0$ is the product order on width-lengths.*

Algorithm 1 Dominant-paths non-restarting vectoring protocol. Node u receives set B of attributes from v pertaining to destination t.

1: $DomTab_u[v, t] := \{a[uv] \oplus b \mid b \in B\}$
2: $Dom_u[t] := \mathcal{D}_{\preceq}(\{DomTab_u[v, t] \mid v \text{ an out-neighbor}\})$
3: **if** $Dom_u[t]$ has changed **then**
4: **for all** r an in-neighbor **do**
5: send $Dom_u[t]$ to r

Routing different flows concurrently on shortest-widest paths, K-quickest paths for all K, and W-wide-shortest paths for all W, is possible by instantiating the dominant-paths vectoring protocols presented in Section 4 with the product order on width-lengths.

4 PROTOCOLS FOR DOMINANT PATHS

We design vectoring protocols that compute on a partial order. If isotonicity is satisfied, then these protocols are able to route on dominant paths. Section 4.1 presents the class of dominant-paths non-restarting vectoring protocols and Section 4.2 presents the class of dominant-paths restarting vectoring protocols.

4.1 Non-restarting protocol

In a standard *non-restarting vectoring protocol*, the destination initiates the routing computation only once, by advertising the neutral attribute to all its in-neighbors, while each node maintains candidate attributes to reach the destination via each of its out-neighbors. At any given moment in time, the node elects the most preferred attribute over all candidate attributes and forwards data-packets to the out-neighbor from which the elected attribute was learned. If the link to that out-neighbor subsequently fails, then the node re-elects a new attribute from among the remaining candidate attributes. Non-restarting vectoring protocols are most common in wired networks, with EIGRP [34] and BGP [33] being prototypical.

Non-restarting vectoring protocols can be generalized to work with a partial order $\preceq$ on attributes. Let $\mathcal{D}_{\preceq}(A)$ denote the set of dominant attributes of set A, which consists of those attributes in A with no attribute in A preferred to them:

$$\mathcal{D}_{\preceq}(A) = \{a \in A \mid \text{there is no } x \in A \text{ such that } x \prec a\}.$$

In the canonical dominant-paths non-restarting vectoring protocol, destination t originates singleton $\{\epsilon\}$, which it advertises to all its in-neighbors. Algorithm 1 presents the pseudo-code for when node u, $u \neq t$, receives a set B of attributes advertised by its out-neighbor v pertaining to destination t. Variable $DomTab_u[v, t]$ stores the set of candidate attributes to reach t via out-neighbor v and variable $Dom_u[t]$ stores the set of elected attributes to reach t.

When u receives set B from v, it first computes the set of attributes learned from v, where each such attribute results from the extension of the attribute of the link to v with an attribute contained in B (line 1). Then, u finds its own new set of elected attributes as the set of dominant attributes from among all candidate attributes learned from each of its out-neighbors (line 2). If there is a change in the set of elected attributes, then u advertises this set to all its in-neighbors (lines 3–5).

Each node assigns a unique label to each of its elected attributes that is advertised to in-neighbors alongside the attribute [8]. Therefore, for a given destination, each node maintains a table with entries of the form

$$attribute\ ,\ label\ :\ next.hop\ ,\ next.label.$$

The table is used as follows:

- Data-packets sourced at the node that need to travel along a path with attribute *attribute*, presumably an optimal path according to some optimality criterion, are forwarded to out-neighbor *next.hop* with label *next.label*.
- Data-packets arriving at the node from an in-neighbor carrying label *label* are forwarded to out-neighbor *next.hop* with the label modified to *next.label*.

A node may install multiple entries with common values of *attribute* and *label*, and different values of *next.hop*. This allows for routing data-packets along multiple dominant paths with a common attribute, a possibility that in standard vectoring protocols is known as ECMP (Equal Cost Multi-Path).

Termination and dominance. A *stable state* of the dominant-paths non-restarting vectoring protocol is a state without advertisements in transit in any of the links of the network. The dominant-paths non-restarting vectoring protocol *terminates* if, in the absence of changes in the network, it reaches a stable state from any initial state.

Inflation alone does not guarantee termination. Two other algebraic properties must be satisfied with respect to the network on which the protocol is run. First, the attribute of every circuit in the network must be strictly inflationary. An attribute a is *strictly inflationary* if $b \prec a \oplus b$ for every non-null attribute b. Widths, extended with min and ordered by $\geq$, provide a simple example where strict inflations fails: whatever width w, it is not the case that $w' > \min(w, w')$ for every width w'; in particular, $w' > \min(w, w')$ is false for $w' \leq w$. We say that a circuit is strictly inflationary if its attribute is strictly inflationary. Inflation together with strictly inflationary circuits prevent oscillatory behaviors.

Second, the set of all path attributes must be finite, to prevent count-to-infinity. This finiteness can ensured by including a hop-count field in attributes and invalidating paths with hop-count in excess of some prespecified maximum value, or, with more precision, by including a field in every attribute that records the path traversed by the advertisements that led up to the attribute and invalidating looping advertisements, as in BGP.

We have the following theorem.

Theorem 4.1. *If all circuits are strictly inflationary and the set of path attributes is finite, then the dominant-paths non-restarting vectoring protocol terminates.*

The proof, which does not rely on isotonicity, is given in Appendix A. It generalizes to partial orders a cognate proof constructed for total orders [36].

Whether or not isotonicity holds, in stable state the dominant-paths non-restarting vectoring protocol routes data-packets via label-switching on paths whose attributes are those elected at the nodes. If isotonicity holds, then these attributes are dominant.

Theorem 4.2. *If isotonicity holds and all circuits are strictly inflationary, then the attributes elected at a node in stable state to reach a destination are the dominant attributes from the node to the destination, that is, $Dom_u[t] = A^*(u,t)$ in stable state for all nodes u and t.*

The proof is given in Appendix B.

Algorithm 2 Dominant-paths restarting vectoring protocol. Node u receives pair (b, n) from v pertaining to destination t.

```
1: att := a[uv] ⊕ b
2: if seq_u[t] < n then
3:     Dom_u[t] := {att}
4:     seq_u[t] := n
5: else if seq_u[t] = n then
6:     Dom_u[t] := D_≤(Dom_u[t] ∪ {att})
7: if Dom_u[t] or seq_u[t] have changed then
8:     for all r an in-neighbor do
9:         send (att, seq_u[t]) to r
```

4.2 Restarting protocol

Contrary to a non-restarting vectoring protocol, in a standard *restarting vectoring protocol*, the destination regularly initiates fresh computation instances during which newly elected attributes replace those from older instances. A node does not maintain candidate attributes to reach the destination via each of its out-neighbors. Rather, it maintains only an elected attribute to reach the destination, which is the most preferred attribute learned so far during the current computation instance. Data-packets are forwarded to the out-neighbor from which the elected attribute was learned. If the link to that out-neighbor fails, the node becomes a traffic black hole for the destination until it learns an attribute coming from a more recent computation instance. The node propagates information about the failure to upstream nodes, so that they too become traffic black holes for a short period of time. DSDV [30] and Babel [11] are restarting protocols proposed for wireless networks. Recently, the interest in restarting vectoring protocols has been renewed in the context of programming routing protocols directly in switching hardware [21, 29], where memory is scarce and instructions are primitive, but computation is plenty and fast.

Restarting vectoring protocols can also be generalized to work with partial orders. In the canonical dominant-paths restarting vectoring protocol, destination t initiates a routing computation instance by advertising attribute ϵ to all its in-neighbors. Algorithm 2 presents the pseudo-code for when node u, $u \neq t$, receives a pair (b, n) from out-neighbor v pertaining to destination t. The pair consists of an attribute b and a sequence number n that identifies the computation instance to which b belongs. Variable $Dom_u[t]$ stores the set of elected attributes to reach t and $seq_u[t]$ holds the sequence number of the computation instance that produces $Dom_u[t]$.

When u receives pair (b, n) from v, it first computes the extension of the attribute of its link to v with b (line 1). If the received attribute is from a computation instance with a higher sequence number than that of the set of elected attributes, then u replaces this set by the singleton consisting of the attribute learned from v and updates its sequence number (lines 2–4). Otherwise, if the learned attribute is from the same computation instance as that of the set of elected attributes and is either preferred to an attribute from this set or is

incomparable with every attribute from the set, then it is included in the set of elected attributes while all less preferred attributes are removed from the set (lines 5–6). The attribute learned from v is advertised to all in-neighbors of u if it has been included in the set of elected attributes or the sequence number has been updated.

As in the non-restarting vectoring protocol, in the restarting vectoring protocol each node maintains, for a given destination, a table with entries of the form

$$attribute\ ,\ label\ :\ next.hop\ ,\ next.label,$$

with the same meaning as before. However, we opt for a version of the protocol with a single entry per value of *attribute*, whose corresponding value of $next.hop$ is the out-neighbor from which *attribute* was first learned during the current computation instance. This option prevents a source from routing data-packets to a destination along multiple dominant paths with a common attribute, but, on the other hand, it guarantees that data-packets travel along loop-free paths without requiring that network circuits be strictly inflationary. Thus, for example, this version of the restarting vectoring protocol can be used to route on performance metrics represented exclusively by widths.

Termination and dominance The concept of termination applies to a single computation instance, which is initiated by a destination when it advertises attribute ϵ to all its in-neighbors. This concept is operationally relevant when the period with which the destination initiates fresh computation instances is large compared to the time it takes for each of them to terminate.

If isotonicity holds, then a restarting vectoring protocol terminates in a stable state where nodes elect dominant attributes to reach destinations. Contrary to non-restarting vectoring protocols, in restarting vectoring protocols strict inflation of circuits and finiteness of path attributes are not necessary for termination and dominance, but, on the other hand, isotonicity must hold even for basic delivery of data-packets. Without isotonicity, these protocols may leave some nodes permanently black holed [37].

5 SERVICE CHAINING CONSTRAINTS

The possibility of designing vectoring protocols that compute on partial orders allows for the solution of routing problems other than those framed exclusively in terms of optimal paths. We illustrate this potential with service chaining. A service chain is a sequence of services that must be applied to a flow of data-packets, each such service being offered by at least one node in the network. In the problem of routing on multiple optimal path criteria constrained by a service chain, we seek to route data-packets on the optimal paths of the various criteria among those that provide the services of the chain in due sequence [6, 10, 21, 31]. This problem can be solved by describing the service chaining constraints in algebraic terms, a process that calls for a partial order.

Since a vectoring protocol computes attributes of paths in the direction from destination to source, attributes encode the *suffix* of the chain completed along a path. For example, the service chain represented by string AB has three suffixes:

- ε, describing a path that does not provide service B;
- B, describing a path that provides service B but not preceded by service A;
- and AB, describing a path that provides service A followed by service B.

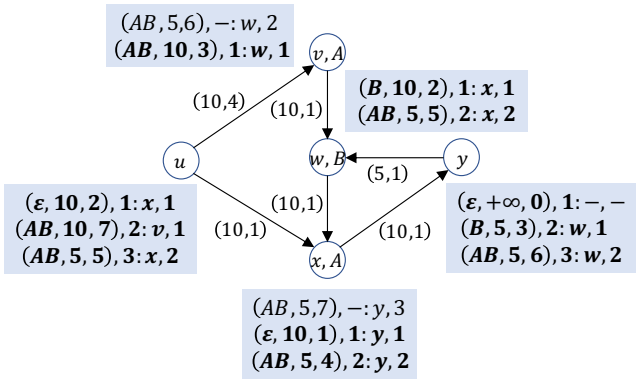


Figure 4: Stable state of a dominant-paths vectoring protocol that routes optimally through service chain AB to destination y. Links are annotated with width-lengths. Nodes v and x offer service A and w offers service B.

The suffix of a chain extends as its services are found along a path. Different suffixes are incomparable.

Figure 4 provides an example of routing on multiple optimality criteria constrained by service chaining. Links are characterized by pairs width-length. Link yw has width 5 and all other links have width 10. Link uv has length 4 and all other links have unit length. The service chain is AB, composed of service A, offered by v and x, followed by service B, offered only by w.

Path uxy, width-length $(10, 2)$, is the only dominant path from u to y without constraints. Paths $uvwxy$, width-length $(10, 7)$, and $uxywxy$, width-length $(5, 5)$, are the two dominant paths from u to y that satisfy service chain constraint AB, the latter path containing cycle $ywxy$. For instance, if data-packets need to be routed along a shortest-widest path subject to service chain AB, then they must be guided along path $uvwxy$, whereas if they need to be routed along a widest-shortest path subject to service chain AB, then they must be guided along path $uxywxy$.

The dominant-paths vectoring protocol computes on triples (X, w, l), where X is a suffix of AB, w is a width and l is a length. Destination y initiates the routing computation with $(\varepsilon, +\infty, 0)$. Node x learns $(\varepsilon, 10, 1)$ from y and u learns $(\varepsilon, 10, 2)$ from x. Node w offers service B. Hence, it learns $(B, 10, 2)$ from x. As a consequence, y learns $(B, 5, 3)$ from w, which is incomparable with the initial triple $(\varepsilon, +\infty, 0)$, because B is incomparable with ε. Both triplets are elected at y. Node x offers service A. Thus, it learns $(AB, 5, 4)$ from $(B, 5, 3)$ elected at y. Triples $(\varepsilon, 10, 1)$ and $(AB, 5, 4)$ are incomparable, since ε and AB are incomparable. Node w further learns $(AB, 5, 5)$ from x and y learns $(AB, 5, 6)$ from w. Node x learns $(AB, 5, 7)$ from y. However, $(AB, 5, 7)$ is less preferred than $(AB, 5, 4)$ and, thus, is not elected. Now, v offers service A. It extends the triples $(B, 10, 2)$ and $(AB, 5, 5)$ elected at w into $(AB, 10, 3)$ and $(AB, 5, 6)$, respectively; the latter of these is less preferred than the former and, thus, it is not elected. Finally, u learns $(AB, 10, 7)$ from v and $(AB, 5, 5)$ from x, in addition to $(\varepsilon, 10, 2)$ previously learned from x.

Labels, which are advertised alongside attributes, enable expedition of data-packets along service-chain constrained optimal paths. For example, if u wants to send data-packets to y along a shortest-widest path subject to chain AB, then it labels them with 1 and forwards them to v; data-packets traverse path $uvwxy$ all the way keeping label 1. If, instead, u wants to send data-packets along a widest-shortest path, also subject to chain AB, then it labels data-packets with 2 and forwards them to x; data-packets traverse path $uxywxy$, changing labels from 2 to 1 on their first passage through y.

6 EVALUATION

Our evaluation intends to answer two main questions. How big are the sets of dominant attributes in realistic networks? Section 6.2 addresses this question. How do dominant-paths vectoring protocols behave during periods of convergence following a network event? Sections 6.3 and 6.4 address this question, respectively, for non-restarting and restarting vectoring protocols. First, Section 6.1 presents the simulator and test networks used in the evaluation.

6.1 Networks and simulator

The test networks used in the evaluation consist of the largest biconnected components of the ISP topologies inferred by the Rocketfuel project [39]. Every link in a topology is annotated with both an OSPF weight and a propagation delay. A width was assigned to each link that is equal to the inverse of its weight, since, by default, OSPF weights are set as inverse capacities; a length was assigned to each link that is equal to its propagation delay. Table 1 characterizes the six networks studied by their numbers of nodes and links.

We built a simulator of vectoring protocols for four instantiations of attributes: pairs width-length, pairs hops-length, pairs width-hops, and triples width-hops-length.[4] Widths extend with the minimum operator, while lengths and hops extend with addition. Each link corresponds to one hop; hence, hops counts the number of links in a path. The product orders on the pairs and the triple were considered as well as several total orders on width-lengths. Advertisements traverse every link first in, first out subject to a random delay taken from a uniform distribution. We set the range of random delays from 0.01 to 1 ms. As a measure against count-to-infinity, advertisements of the non-restarting vectoring protocols that travel more than a prespecified maximum number of hops are invalidated. We set that maximum number to 15 as in RIP.

6.2 Sets of dominant attributes

The number of dominant attributes from sources to destinations determine the viability of our approach to routing on multiple optimality criteria. These sets can be read off from the stable state of a dominant-paths vectoring protocol. Table 1 shows the average number of dominant width-lengths over all source-destination pairs for all six networks. This number varies between 1.9 and 3.7. In every network, the number of distinct path widths is an upper bound on the number of dominant width-lengths for all source-destination pairs. On the other hand, the number of distinct path widths in a network equals the number of distinct link widths, since the width of a path is the width of one of its links. Table 1 shows that the average number of dominant width-lengths in the test networks is well below the upper bound of the number of distinct link widths.

[4]The source code of the simulator is available at https://github.com/miferrei/rmoc-sigcomm2020-artifact.

Table 1: Number of nodes, number of links, number of distinct widths, and number of dominant width-lengths for the six networks considered.

AS number	Nodes	Links	Distinct widths	Dominant width-lengths
1221	50	194	8	1.9
1239	284	1882	19	2.5
1755	73	292	18	2.2
3257	113	558	21	3.5
3967	72	280	19	3.7
6461	129	726	19	2.8

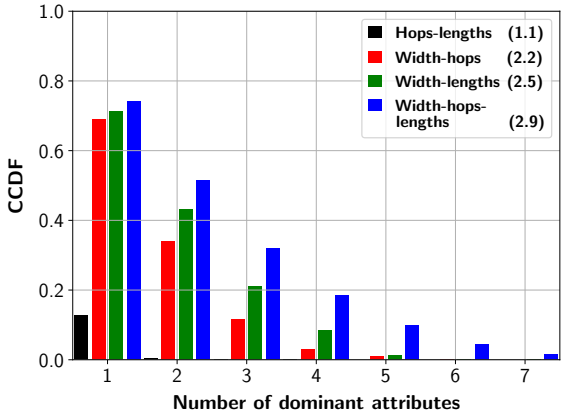
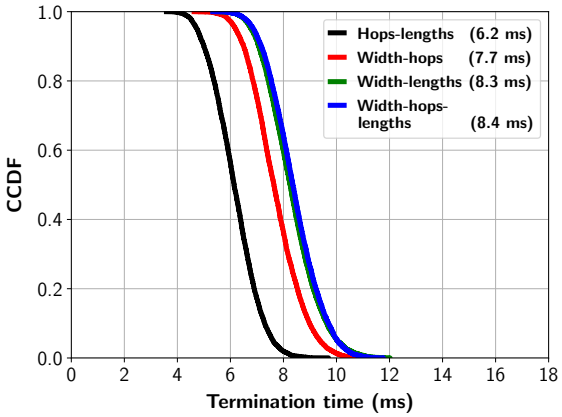
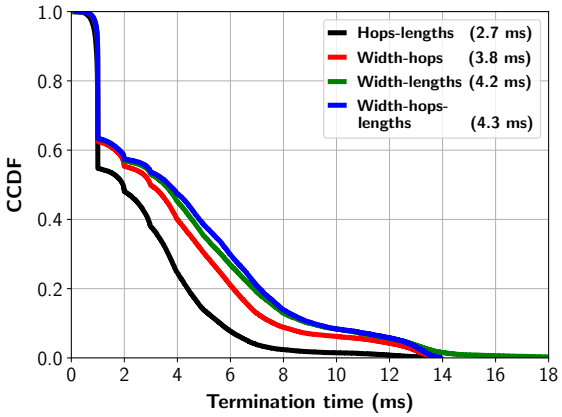


Figure 5: CCDF of the number of dominant attributes in AS 1239 for the product orders on hops-lengths, width-hops, width-lengths, and width-hops-lengths. Averages of the distributions are given inside parenthesis.

Figure 5 shows the Complementary Cumulative Distribution Function (CCDF) of the number of dominant attributes for the largest of the networks, AS 1239, and all four instantiations of attributes. Path lengths are highly correlated with path hops. In other words, in most cases a shortest path contains a minimum number of links. The average number of dominant hops-lengths is only 1.1, while the maximum such number is three. The percentages of source-destination pairs connected by more than three dominant attributes are 11.6%, 21.2%, and 31.9%, respectively for width-hops, width-lengths, and width-hops-lengths, while the averages are 2.2, 2.5, and 2.9. There is more diversity in path lengths than in path hops, which justifies the finding that there are more dominant width-lengths than dominant width-hops from sources to destinations. Dominant triples width-hops-length contain dominant pairs hops-length, width-hops, and width-length. Thus, sets of dominant width-hops-lengths are expected to be larger than the other sets of dominant attributes.



(a) Network-wide announcement of a destination.



(b) Failure of a link.

Figure 6: CCDFs of termination times in AS 1239 for a dominant-paths non-restarting vectoring protocol operating on the product orders on hops-lengths, width-hops, width-lengths, and width-hops-lengths. Averages of the distributions are given inside parenthesis.

6.3 Transient behavior of non-restarting protocols

We assess the transient behavior of non-restarting vectoring protocols in AS 1239 against two types of network events: the network-wide announcement of a destination and the failure of a link. The metric used is the *termination time*, defined as the duration of the interval of time elapsed from the moment a network event occurs until the protocol reaches a stable state. For each network event, we ran 25 independent trials.

Announcement of a destination. Figure 6a shows the CCDF of the termination time after a network-wide announcement over all possible destination nodes and all trials. These curves are rather smooth and steep (small variance) for all instantiations of attributes. The average termination times are 6.2 ms, 7.7 ms, 8.3 ms, and 8.4 ms, respectively for hops-lengths, width-hops, width-lengths, and width-hops-lengths (the curves for width-lengths and width-hops-lengths almost coincide).

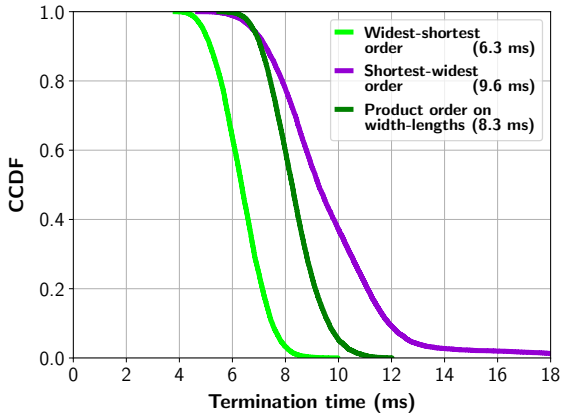
Because of isotonicity, elected attributes at each node to reach a destination can only be replaced by more preferred attributes during each trial. Therefore, the termination time equals the maximum delay to propagate an advertisement all the way up a dominant path and is, thus, roughly proportional to the number of links in a dominant path with the largest such number. As observed before, path lengths and path hops are correlated. On the other hand, path widths and path hops are not necessarily correlated; wide paths from source to destination typically traverse more than the minimum number of links required to reach the destination from the source. This justifies the observed fact that those attributes that involve width lead to longer termination times.

Failure of a link. Figure 6b shows the CCDF of the termination time after a link failure over all possible links and all trials. The nodes of AS 1239 are clustered around geographical areas, with nodes inside each cluster densely connected with links of unit length. A failure of one of these links has only a localized impact on state of the protocol. As observed from the figure, 45.3%, 37.4%, 36.7%, and 36.6% of the failures have termination times equal or less than 1 ms, respectively for hops-lengths, width-hops, width-lengths, and width-hops-lengths.

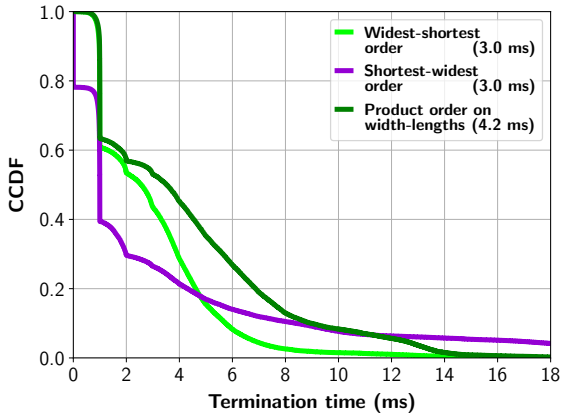
Apart from this effect, the CCDFs of termination time for a link failure are more long tailed (large variance) than those for a network-wide announcement of a destination. For example, 1.5%, 6.2%, 8.4%, and 8.5% of the failures lead to termination times in excess of 10 ms, respectively for hops-lengths, width-hops, width-lengths, and width-hops-lengths. In a non-restarting vectoring protocol, all nodes continuously search for dominant attributes over all candidate attributes learned from its out-neighbors, while a link failure ultimately requires some nodes to stabilize on less preferred attributes than those they started out with. Trying to settle on less preferred attributes by always electing dominant attributes among candidates learned from out-neighbors takes many iterations, corresponding to as many paths being explored and, hence, to long termination times [23].

Comparison against various optimality criteria. Figure 7a shows the CCDF of the termination time after a network-wide announcement in AS 1239 over all possible destination nodes and all trials. We plot curves for a dominant-paths vectoring protocol operating on the product order on width-lengths against two instantiations of a standard vectoring protocol: one operating according to the widest-shortest order and the other operating according to the shortest-widest order.

Two important conclusions emerge from Figure 7a. First, the termination time of the dominant-paths vectoring protocol is not too far off from the termination time of a standard vectoring protocol for widest-shortest paths. The respective averages are 8.3 ms and 6.3 ms. This is somewhat surprising given that the standard vectoring protocol elects only one width-length per node per destination, whereas the dominant-paths vectoring protocol elects, on average, 2.5 width-lengths per node per destination, with 21.2% of the nodes electing more than three width-lengths per destination (see Figure 5). The explanation for the favorable termination time of a dominant-paths vectoring protocol is that the multiple



(a) Network-wide announcement of a destination.



(b) Failure of a link.

Figure 7: CCDFs of termination times in AS 1239 for a dominant-paths non-restarting vectoring protocol operating on the product order on width-lengths and for a standard non-restarting vectoring protocol operating on the widest-shortest order and on the shortest-widest order. Averages of the distributions are given inside parenthesis.

attributes comprising a set of dominant attributes are elected in parallel during an execution of the protocol.

Second, the termination time of the dominant-paths vectoring protocol is better than the termination time of a standard vectoring protocol for shortest-widest paths. The respective averages are 8.3 ms and 9.6 ms, with the CCDF of the termination time of the latter protocol having a long tail. The explanation lies on isotonicity, which promotes fast convergence. The shortest-widest order does not satisfy isotonicity, implying that following a network-wide announcement of a destination, a node may elect a width-length that later has to be supplanted by a less preferred width-length. As mentioned before in the case of a link failure, such a process is slow. The standard vectoring protocol for shortest-widest paths not only takes longer to terminate, but, we recall, in stable state may not route data-packets on shortest-widest paths (see Section 2.1).

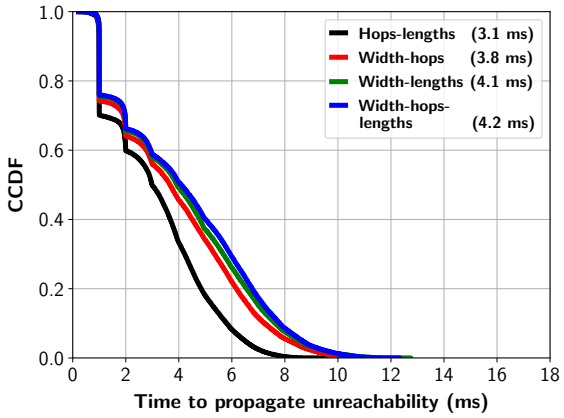


Figure 8: CCDFs of the time to propagate unreachability in AS 1239 for a dominant-paths restarting vectoring protocol operating on the product orders on hops-lengths, width-hops, width-lengths, and width-hops-lengths. Averages of the distributions are given inside parenthesis.

In the case of AS 1239, the protocol does not route data-packets on shortest-widest paths for 30.3% of source-destination pairs.

Figure 7b shows the CCDF of the termination time after a link failure in AS 1239 over all possible links and all trials. For the cases of widest-shortest paths and dominant width-length paths, all link failures cause some alteration in the stable state of the protocols. However, for the case of shortest-widest paths 21.8% of link failures do not affect the stable state of the protocol. When the set of dominant paths contains paths with minimum number of links, the failure of a link will prevent the node at its tail from communicating directly with the node at its head. Therefore, at least that node will need to update its stable state. We previously observed that path lengths are strongly correlated with paths hops, which explains why all link failures cause some alteration in the case of widest-shortest paths and dominant width-length paths. However, shortest-widest paths favor large widths independently of the number of links in a path, implying that some links may not be selected for routing. The failure of such links does not change the state of the protocol.

Last, we observe that, as in the case of a network-wide announcement of a destination, the curve for shortest-widest paths exhibits the longest tail.

6.4 Transient behavior of restarting protocols

Every computation instance initiated by a destination in a dominant-paths restarting vectoring protocol behaves as a network-wide announcement of a destination in a non-restarting vectoring protocol. Hence, it can be characterized by the termination times shown in Figure 6a. When a link fails, the node directly affected by the failure propagates unreachability information to its upstream nodes. Figure 8 shows the CCDF of the time it takes for all nodes to withdraw the elected attributes corresponding to paths that contain the failed link over all links and all trials. This time duration is smaller than the termination time after a network-wide announcement

of a destination because only a fraction of nodes are affected by the failure.

While unreachability information propagates upstream the failure, some nodes may be momentarily traffic black holed. Our simulations show that reachability from source to destination is interrupted, on average, on only 0.1% of the cases for width-lengths, width-hops, and width-hops-lengths, and only on 0.2% of the cases for hops-lengths. The duration of the interruptions decreases with the period between consecutive computation instances initiated by the destination.

7 RELATED WORK

Algebraic conceptualization of routing. The algebraic framework proposed in References [18, 35, 36] laid the foundations for a unified treatment of routing problems and protocols, abstracting away the specificity of performance metrics and protocol parameters. The framework is premised on a total order on attributes. The present work generalizes the algebraic framework by accepting partial orders on attributes and by devising vectoring protocols that compute on them. Moreover, it describes a generic procedure that reduces a set of total orders to a common partial order that is isotone and respects all total orders.

Multi-objective path problems. Multi-objective path problems have been studied by the operations research community [5, 19, 28]. These problems can be described in concrete algebraic terms. Attributes are tuples of the Cartesian product of elementary metrics, each of which either extends with + and is totally ordered by $\leq$ or extends with min and is totally ordered by $\geq$ (or extends with max and is ordered by $\leq$). Tuples extend term-wise and are partially ordered by the product order of their term-wise total orders. The goal is to find sets of dominant tuples from source to destination in a network and is attained with generalizations of Dijkstra's and Bellman-Ford algorithms [5, 19, 28].

The setting considered in this work is broader and the problem addressed is different. Attributes are not necessarily tuples of elementary metrics. Even when they are, a partial order on them is derived, rather than assumed a priori, and does not necessarily coincide with the product order. The goal is to route data-packets on multiple optimality criteria and is attained with dominant-paths vectoring protocols.

Multipath routing protocols. Since dominant-paths vectoring protocols typically find multiple paths from source to destination, they can be considered a type of multipath routing protocols. Multipath routing protocols have mostly been proposed as extensions to BGP with one of the following three goals in mind. A first goal is to ensure termination both of external BGP [1] and of internal BGP [13, 40]. A second goal is to improve the data-packet delivery capabilities of BGP during convergence of the protocol upon a link failure [14, 22, 27, 41]. And a third goal is to allow the configuration of more expressive routing policies than is possible with standard BGP [44]. The BGP multipath routing proposal presented in [42] addresses all three goals.

The dominant-paths vectoring protocols proposed in this work target a different goal. We seek to route data-packets on optimal paths for a variety of optimality criteria, some of which do not lend themselves to a solution by a standard vectoring protocol. In addition, our dominant-paths vectoring protocols are formulated with generality rather than being specific to BGP.

Regular-expression-constrained routing. The service chaining constraints on routing discussed in the text are a particular case of constraints dictated by regular expressions [26] on annotations provided to network links. The problem of finding a shortest path in a network subject to such a constraint is defined in Reference [3] and mapped to the problem of finding a (unconstrained) shortest path in a special product graph that combines the network and the automaton describing the regular expression. The application of this idea to routing envisages a pre-computation of the product graph. In References [4, 32, 38], the shortest paths on the product graph are centrally computed, whereas in Reference [21], they are computed by a routing protocol running on the nodes of the product graph.

The primary problem addressed in this work is routing on multiple optimality criteria. The solution to this problem leads to dominant-paths vectoring protocols that compute on partially ordered sets of attributes. By modeling path constraints within a set of attributes, the protocols provide a fully distributed solution to the problem of constrained optimal path routing. Although the text only considered the case of service chaining, the generalization to an arbitrary regular expression should not pose major conceptual difficulties.

8 CONCLUSIONS AND DISCUSSION

We presented a solution to the problem of routing on multiple optimality criteria based on the ideas of: (1) intersecting the total orders of all criteria; and (2) reducing the resulting intersection to satisfy isotonicity. This process leads to partial orders. We designed vectoring protocols that compute on partial orders to find dominant attributes and paths from sources to destinations in any given network. Alongside the advertisement of routing information, these protocols disseminate the necessary forwarding information to guide data-packets on dominant paths. Preliminary evaluations indicate that these protocols converge fast and elect only a few attributes at each node to reach a destination. While our working examples emphasized widths, lengths, and hop-counts, the ideas were developed for arbitrary metrics that satisfy the algebraic properties of associativity, commutativity, and inflation. Further bolstering the generality of our approach, we showed how service chaining constraints could be formulated in terms of partial orders.

To a large extent, the concepts presented here apply to link-state routing protocols and to centralized control planes. In these cases, dominant attributes and paths are computed by a sequential algorithm. A dominant-paths version of Dijkstra's algorithm can be designed for this purpose. Then, some forwarding mechanism must be put in place to guide data-packets on dominant paths.

ACKNOWLEDGMENTS

We thank José Brázio and Matthew Roughan, our shepherd, for illuminating discussions positively reflected in the paper. We thank Nuno Lopes, Fernando Ramos, and the anonymous reviewers for the many constructive comments they provided. This work was partially funded by Portuguese Fundação para a Ciência e Tecnologia under grant UIDB/50008/2020.

REFERENCES

[1] Rachit Agarwal, Virajith Jalaparti, Matthew Caesar, and P. Brighten Godfrey. 2010. Guaranteeing BGP Stability with a Few Extra Paths. In *Proc. of the IEEE International Conference on Distributed Computing Systems*. 221–230.

[2] Toufik Ahmed, Ahmed Mehaoua, Raouf Boutaba, and Youssef Iraqi. 2005. Adaptive Packet Video Streaming over IP Networks: a Cross-layer approach. *IEEE Journal on Selected Areas in Communications* 23, 2 (2005), 385–401.

[3] Christopher L. Barrett, Riko Jacob, and Madhav V. Marathe. 2000. Formal-Language-Constrained Path Problems. *SIAM J. Comput.* 30, 3 (2000), 809–837.

[4] Ryan Beckett, Ratul Mahajan, Todd Millstein, Jitendra Padhye, and David Walker. 2016. Don't Mind the Gap: Bridging Network-wide Objectives and Device-level Configurations. In *Proc. ACM SIGCOMM*. 328–341.

[5] James Brumbaugh-Smith and Donald Shier. 1989. An Empirical Investigation of Some Bicriterion Shortest Path Algorithms. *European Journal of Operational Research* 43, 2 (1989), 216–224.

[6] Zizhong Cao, Murali Kodialam, and T. V. Lakshman. 2014. Traffic Steering in Software Defined Networks: Planning and Online Routing. In *Proc. of the ACM SIGCOMM Workshop on Distributed Cloud Computing*. 65–70.

[7] Bernard Carré. 1979. *Graphs and Networks*. Clarendon Press, Oxford, UK. ISBN 0-19-8596-22-7.

[8] Girish P. Chandranmenon and George Varghese. 1996. Trading Packet Headers for Packet Processing. *IEEE/ACM Transactions on Networking* 4, 2 (1996), 141–152.

[9] Yen L. Chen and Yeo H. Chin. 1990. The Quickest Path Problem. *Computers & Operation Research* 17 (1990), 153–161.

[10] Sumi Choi, Jonathan Turner, and Tilman Wolf. 2001. Configuring Sessions in Programmable Networks. In *Proc. IEEE INFOCOM*. 60–66.

[11] Juliusz Chroboczek. 2011. *The Babel Routing Protocol*. RFC 6126.

[12] Thomas Cormen, Charles Leiserson, Ronald Rivest, and Clifford Stein. 2009. *Introduction to Algorithms* (third ed.). MIT Press, Cambridge, MA. ISBN 978-0262033848.

[13] Ashley Flavel and Matthew Roughan. 2009. Stable and Flexible iBGP. In *Proc. ACM SIGCOMM*. 183–194.

[14] Igor Ganichev, Bin Dai, Philip B. Godfrey, and Scott Schenker. 2010. YAMR: Yet Another Multipath Routing Protocol. *ACM SIGCOMM Computer Communications Review* 5 (October 2010), 13–19.

[15] Michel Gondran and Michel Minoux. 2008. *Graphes, Dioides, and Semirings*. Springer. ISBN 978-0-387-75449-9.

[16] Mohamed G. Gouda and Marco Schneider. 2003. Maximizable Routing Metrics. *IEEE/ACM Transactions on Networking* 11, 4 (2003).

[17] Timothy G. Griffin. 2010. The Stratified Shortest-paths Problem. In *Proc. International Conference on Comunication Systems and Networks*. 268–277.

[18] Timothy G. Griffin and João L. Sobrinho. 2005. Metarouting. In *Proc. ACM SIGCOMM*. 1–12.

[19] Pierre Hansen. 1980. Bicriterion Path Problems. In *Multiple Criteria Decision Making Theory and Application*, Gunter Fandel and Tomas Gal (Eds.). Springer Verlag, 109–127.

[20] Egbert Harzheim. 2005. *Ordered Sets*. Springer. ISBN 0-387-24219-8.

[21] Kuo-Feng Hsu, Ryan Beckett, Ang Chen, Jennifer Rexford, Praveen Tammana, and David Walker. 2020. Contra: A Programmable System for Performance-Aware Routing. In *Proc. USENIX NSDI*.

[22] Nate Kushman, Srikanth Kandula, Dina Katabi, and Bruce M. Maggs. 2007. R-BGP: Staying Connected in a Connected World. In *Proc. USENIX NSDI*.

[23] Craig Labovitz, Abha Ahuja, Abhijit Bose, and Farnam Jahanian. 2001. Delayed Internet Routing Convergence. *IEEE/ACM Transactions on Networking* 9, 3 (2001), 293–306.

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

[25] Thomas Lengauer and Dirk Theune. 1991. Efficient Algorithms for Path Problems with General Cost Criteria. In *Proc. International Colloquium on Automata, Languages and Programming*. 314–326.

[26] Harry Lewis and Christos Papadimitriou. 1998. *Elements of the Theory of Computation* (second ed.). Prentice-Hall. ISBN 0132624788.

[27] Yong Liao, Lixin Gao, Roch Guérin, and Zhi-Li Zhang. 2008. Reliable Interdomain Routing Through Multiple Complementary Routing Processes. In *Proc. ACM CoNEXT*.

[28] Ernesto Martins. 1984. On a Multicriteria Shortest Path Problem. *European Journal of Operational Research* 16, 2 (1984), 236–245.

[29] Edgar C. Molero, Stefano Vissicchio, and Laurent Vanbever. 2018. Hardware-Accelerated Network Control Planes. In *Proc. ACM Workshop on Hot Topics in Networks*. 120–126.

[30] Charles E. Perkins and Pravin Bhagwat. 1994. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *Proc. ACM SIGCOMM*. 234–244.

[31] Paul Quinn and Thomas D. Nadeau. 2015. Problem Statement for Service Function Chaining. (2015). RFC 7498.

[32] Mark Reitblatt, Marco Canini, Arjun Guha, and Nate Foster. 2013. FatTire: Declarative Fault Tolerance for Software-defined Networks. In *Proc. of the ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*. ACM, 109–114.

[33] Yakov Rekhter, T. Li, and S. Hares. 2006. *A Border Gateway Protocol 4 (BGP-4)*. RFC 4271.

[34] Donnie Savage, James Ng, Steven Moore, Donald Slice, Peter Paluch, and Russ White. 2016. *Cisco's Enhanced Interior Gateway Routing Protocol (EIGRP)*. RFC 7868.

[35] João L. Sobrinho. 2002. Algebra and Algorithms for QoS Path Computation and Hop-by-hop Routing in the Internet. *IEEE/ACM Transactions on Networking* 10, 4 (2002).

[36] João L. Sobrinho. 2005. An Algebraic Theory of Dynamic Network Routing. *IEEE/ACM Transactions on Networking* 13, 5 (2005).

[37] João L. Sobrinho. 2019. Fundamental Differences Among Vectoring Routing Protocols on Non-Isotonic Metrics. *IEEE Networking Letters* 1, 3 (2019).

[38] Robert Soulé, Basu Shrutarshi, Parisa J. Marandi, Fernando Pedone, Robert Kleinberg, Emin Gun Sirer, and Nate Foster. 2014. Merlin: A Language for Provisioning Network Resources. In *Proc. CoNEXT*. 213–226.

[39] Neil Spring, Ratul Mahajan, David Wetherall, and Thomas Anderson. 2004. Measuring ISP Topologies with Rocketfuel. *IEEE/ACM Transactions on Networking* 12, 1 (February 2004), 2–16.

[40] Virginie Van den Schrieck, Pierre Francois, and Olivier Bonaventure. 2010. BGP Add-paths: the Scaling Performance Tradeoffs. *IEEE Journal on Selected Areas in Communications* 28, 8 (2010), 1299–1307.

[41] Feng Wang and Lixin Gao. 2008. A Backup Route Aware Routing Protocol—Fast Recovery From Transient Routing Failures. In *Proc. IEEE INFOCOM*. 2333–2341.

[42] Yi Wang, Michael Schapira, and Jennifer Rexford. 2009. Neighbor-specific BGP: More Flexible Routing Policies While Improving Global Stability. In *Proc. ACM SIGMETRICS*. 217–228.

[43] Zheng Wang and Jon Crowcroft. 1996. Quality-of-Service Routing for Supporting Multimedia Applications. *IEEE Journal on Selected Areas in Communications* 14, 7 (September 1996), 1228–1234.

[44] Wen Xu and Jennifer Rexford. 2006. MIRO: Multi-path Interdomain Routing. In *Proc. ACM SIGCOMM*. 171–182.

[45] Yaling Yang and Jun Wang. 2008. Design Guidelines for Routing Metrics in Multihop Wireless Networks. In *Proc. IEEE INFOCOM*. 1615–1623.

Appendices are supporting material that has not been peer-reviewed.

A TERMINATION OF NON-RESTARTING VECTORING PROTOCOLS

We assume that every circuit in the network is strictly inflationary and that the set of all possible path attributes is finite. We want to show that the dominant-paths non-restarting vectoring protocol terminates. The high-level idea of the proof consists in finding a map from the state of the protocol to a well-ordered set [20][5] such that the value of the map decreases every time a node advertises a set of elected attributes to its in-neighbors. As a well-ordered set does not contain an infinite strictly decreasing sequence, the number of advertisements must be finite, which implies termination of the protocol [24, 36].

We start with a lemma characterizing the set of elected attributes at a node to reach a destination following the reception of a set of attributes.

Lemma A.1. *Suppose that a node receives a set of attributes advertised by one of its out-neighbors. Every attribute newly elected as a consequence of this reception is either the extension of a newly advertised attribute or is less preferred than an attribute elected before the reception, or both.*

Proof. Suppose that node u receives set B of attributes advertised by its out-neighbor v pertaining to some destination. This set contains a set B^* of newly advertised attributes, which are those that were not included in the previous advertisement sent by v

[5] A set is *well-ordered* if every one of its non-empty subsets has a least element in the order.

to u. Let E and E' denote the set of attributes elected at u before and after reception of B, respectively. We want to show that for every $e \in E' - E$, there is $b \in B^*$ such that $e = a[uv] \oplus b$ or there is $f \in E - E'$ such that $f \prec e$.

If e was not a candidate attribute for election at u before B is received, then e must have been learned from an attribute in B^*, that is, there is $b \in B^*$ such that $e = a[uv] \oplus b$. Otherwise, if e was a candidate attribute for election at u before B is received, then there must have existed an elected attribute that prevented the election of e, that is, there is $f \in E$ such that $f \prec e$. Moreover, since e is elected after B is received, f is no longer a candidate attribute for election at u, implying that f is no longer elected, $f \notin E'$. Hence, $f \in E - E'$. □

We now define the *node-attribute digraph* as follows:

Vertices. Vertices are pairs node-attribute (u, a) such that u is a node and a is either a candidate attribute at u or an attribute contained in an advertised set of attributes in transit from u to one of its in-neighbors at any state during execution of the protocol. Pairs (t, ϵ) with t a destination are included. Every node-attribute (u, a) is of the form $(u, a[P] \oplus b)$, where P is a path from u to a node v with (v, b) a node-attribute pair at the initial state. Since we assume that $a[P] = \bullet$ for every path P with sufficiently large number of links, the set of node-attributes is finite.

Arcs. Arcs are of two types: (1) an extension arc $((u, a), (v, b))$ if uv is a link in the network, $b \neq \bullet$, and $a = a[uv] \oplus b$; and (2) a selection arc $((u, a), (u, b))$ if $b \prec a$.

Lemma A.2. *The node-attribute digraph is acyclic.*

Proof. The proof is by contradiction. Let

$$C = (u_0, a_0)(u_1, a_1) \cdots (u_{n-1}, a_{n-1})(u_0, a_0)$$

be a cycle in the node-attribute digraph with minimum number of arcs. Clearly, a_0 is not null, since neither extension arcs nor selection arcs enter $(u_0, \bullet)$. If $((u_i, a_i), (u_{i+1}, a_{i+1}))$ is an extension arc ($0 \leq i < n$ and addition modulus n), then, because of inflation, $a_{i+1} \preceq a[u_i u_{i+1}] \oplus a_{i+1} = a_i$. Otherwise, if $((u_i, a_i), (u_{i+1}, a_{i+1}))$ is a selection arc, then $a_{i+1} \prec a_i$.

Suppose that at least one arc in cycle C is a selection arc. Let $((u_i, a_i), (u_{i+1}, a_{i+1}))$ be one such arc. We have

$$a_0 \preceq a_{n-1} \preceq \cdots \preceq a_{i+1} \prec a_i \preceq \cdots \preceq a_1 \preceq a_0,$$

which is a false statement. Instead, suppose that no arc in C is a selection arc. Then, all arcs in C are extension arcs. Since C was chosen with minimum number of arcs, $u_0 u_1 \cdots u_{n-1} u_0$ is a circuit in the network that starts and ends at u_0. We have

$$\begin{aligned} a_0 &= a[u_0 u_1] \oplus \cdots \oplus a[u_{n-1} u_0] \oplus a_0 \\ &= a[u_0 u_1 \cdots u_{n-1} u_0] \oplus a_0, \end{aligned}$$

which contradicts the fact that attribute $a[u_0 u_1 \cdots u_{n-1} u_0]$, being the attribute of a circuit, is strictly inflationary. □

Theorem A.3. *If all circuits are strictly inflationary and the set of path attributes is finite, then the dominant-paths non-restarting vectoring protocol terminates.*

Proof. From Lemma A.2, node-attributes can be topologically ordered [12] such that for every arc $((u, a), (v, b))$ node-attribute (v, b) is topologically smaller than node-attribute (u, a). Denote by N the number of node-attribute pairs.

Let Δ be the set of N-tuples of nonnegative integers with terms indexed by the topologically ordered set of node-attributes. Set Δ is lexicographically ordered. Given $\alpha, \beta \in \Delta$, α is lexicographically smaller than β if there is a node-attribute (u, a) such that $\alpha_{(u,a)} < \beta_{(u,a)}$ and $\alpha_{(v,b)} = \beta_{(v,b)}$ for all node-attributes (v, b) that are topologically smaller than (u, a). The lexicographic order well-orders Δ.

We present a map Γ from the state of the protocol to Δ whose value decreases lexicografically every time a node advertises a set of attributes to its in-neighbors. The value of the term of Γ with index (u, a), $\Gamma_{(u,a)}$, is defined by

- number of times node u elects attribute a plus the number of times attribute a is newly advertised in sets of attributes in transit from u to its in-neighbors, over all destinations.

Suppose that u receives set B of attributes advertised by its out-neighbor v pertaining to some destination. Set B contains set B^* of newly advertised attributes. Let E and E' denote the set of attributes elected at u before and after reception of B, respectively. We have:

- for every $b \in B^*$, the value of $\Gamma_{(v,b)}$ decreases by one;
- for every $f \in E - E'$, the value of $\Gamma_{(u,f)}$ decreases by one;
- for every $e \in E' - E$, the value $\Gamma_{(u,e)}$ increases by one plus the number of in-neighbors of u.

Node u advertises set E' of attributes to its in-neighbors if $E' - E$ is not empty or $E - E'$ is not empty. If $E' - E = \emptyset$ and $E - E' \neq \emptyset$, then no value of a term of Γ increases, while for every $f \in E - E'$ the value of $\Gamma_{(u,f)}$ decreases. Hence, the value of Γ decreases lexicographically. On the other hand, if $E' - E \neq \emptyset$, then for every $e \in E' - E$ the value of $\Gamma_{(u,e)}$ increases. Lemma A.1 asserts that for every $e \in E' - E$, there is $b \in B^*$ such that $e = a[uv] \oplus b$ or there is $f \in E - E'$ such that $f \prec e$. Both (v, b) and (u, f) are topologically smaller than (u, e). Therefore, the increase in the value of $\Gamma_{(u,e)}$ is accompanied by a decrease in the value of a term of Γ with a topologically smaller index, $\Gamma_{(v,b)}$ or $\Gamma_{(u,f)}$. Hence, in this case as well, the value of Γ decreases lexicographically. Since Δ is well-ordered by the lexicographic order, Γ cannot decrease indefinitely, implying an end to the advertisement of sets of attributes. □

B DOMINANCE OF NON-RESTARTING VECTORING PROTOCOLS

We assume that binary extension operation $\oplus$ is isotone for partial order $\preceq$ and that every circuit in the network is strictly inflationary. We want to show that the set of elected attributes at node u to reach destination t in stable state, herein denoted by $E(u, t)$, is the set of dominant attributes from u to t, $A^*(u, t)$.

Destination t always elects singleton $\{\epsilon\}$. In stable state, the candidate attributes at u to reach t are extensions of elected attributes at its various out-neighbors. The set of elected attributes at u to reach t satisfies the following fixed-point equation:

$$E(u, t) = \mathcal{D}_{\preceq}(\{a[uv] \oplus b | b \in E(v, t), v \text{ out-neighbor of } u\}).$$

Lemma B.1. *Every dominant attribute from a node to a destination is the attribute of a simple path from the node to the destination.*

Proof. We show that the attribute of a path containing a circuit either equals or is less preferred than the attribute of the path obtained through removal of the circuit.

Let PCQ be a path from node u to destination t that contains circuit C. Because of inflation, we write

$$a[Q] \preceq a[C] \oplus a[Q] = a[CQ].$$

And because of isotonicity, we obtain

$$a[PQ] = a[P] \oplus a[Q] \preceq a[P] \oplus a[CQ] = a[PCQ],$$

showing that the attribute of path PCQ either equals or is less preferred than the attribute of path PQ. □

Lemma B.2. *Every non-null attribute belonging to the set of elected attributes at a node to reach a destination is the attribute of a simple path from the node to the destination.*

Proof. Let a_0 be a non-null attribute elected at node u_0 other than destination t. Attribute a_0 is the extension of some non-null attribute a_1 elected at an out-neighbor u_1 of u_0. In turn, either $u_1 = t$ and $a_1 = \epsilon$, or a_1 is the extension of some attribute a_2 elected at an out-neighbor u_2 of u_1. Continuing this process of moving from a node to one of its out-neighbors through extended attributes, we either arrive at t or at a node visited before. Strict inflation excludes the latter hypothesis, so that a_0 is the attribute of a simple path from u_0 to t.

The argument above is made precise as follows. Suppose we are given $a_0 \in E(u_0, t)$, $a_0 \neq \bullet$, and $u_0 \neq t$. Let $u_0u_1 \cdots u_{n-1}$ be the longest simple path starting at u_0 such that

$$a_i = a[u_iu_{i+1} \cdots u_{n-1}] \oplus a_{n-1} \in E(u_i, t),$$

for all i, $0 \leq i < n-1$, and $a_{n-1} \in E(u_{n-1}, t)$. Either $u_{n-1} = t$ or $u_{n-1} \neq t$. In the former case, $a_{n-1} \in E(t, t) = \{\epsilon\}$ and, thus, $a_0 = a[u_0u_1 \cdots u_{n-1}] \oplus \epsilon = a[u_0u_1 \cdots u_{n-1}]$, showing that a_0 is the attribute of a simple path from u_0 to t.

We now prove that the case $u_{n-1} \neq t$ leads to a contradiction, thereby concluding that a_0 is indeed the attribute of a simple path from u_0 to t. The fixed-point equation for $E(u_{n-1}, t)$ asserts that there is an out-neighbor u_n of u_{n-1} and an attribute $a_n \in E(u_n, t)$ such that $a_{n-1} = a[u_{n-1}u_n] \oplus a_n$. Therefore, we may write

$$\begin{aligned} a_i &= a[u_iu_{i+1} \cdots u_{n-1}] \oplus a_{n-1} \\ &= a[u_iu_{i+1} \cdots u_{n-1}] \oplus a[u_{n-1}u_n] \oplus a_n \\ &= a[u_iu_{i+1} \cdots u_{n-1}u_n] \oplus a_n, \end{aligned}$$

for all i, $0 \leq i < n$, and $a_n \in E(u_n, t)$. Since $u_0u_1 \cdots u_{n-1}$ was the longest simple path starting at u_0 such that $a_i = a[u_iu_{i+1} \cdots u_{n-1}] \oplus a_{n-1}$ for all i, $0 \leq i < n-1$, it must be the case that $u_j = u_n$ for some j, $0 \leq j < n$. Path $u_nu_{j+1} \cdots u_{n-1}u_n$ is a circuit and it satisfies

$$a_j = a[u_nu_{j+1} \cdots u_{n-1}u_n] \oplus a_n.$$

The circuit is strictly inflationary, so that $a_j \prec a_n$, which contradicts the fact that both a_j and a_n belong to $E(u_n, t)$ and, thus, must be incomparable. □

Lemma B.3. *The attribute of every path from a node to a destination either equals or is less preferred than at least one of the attributes elected at the node to reach the destination.*

Proof. Let $u_{n-1} \cdots u_1u_0$ be any path from node $u = u_{n-1}$ to destination $t = u_0$. We prove by induction that there is $a_{n-1} \in E(u_{n-1}, t)$ such that $a_{n-1} \preceq a[u_{n-1} \cdots u_1u_0]$. (The indexing of the nodes of a path from its destination towards its source simplifies the induction proof.)

The base case is the election of ϵ at destination t, which is also the attribute of the trivial path composed of t alone. For the induction step, assume that there is $a_i \in E(u_i, t)$ such that $a_i \preceq a[u_i \cdots u_1u_0]$. From isotonicity, we write

$$a[u_{i+1}u_i] \oplus a_i \preceq a[u_{i+1}u_i] \oplus a[u_i \cdots u_1u_0] = a[u_{i+1} \cdots u_1u_0].$$

The fixed-point equation for u_{i+1} implies that there is $a_{i+1} \in E(u_{i+1}, t)$ such that

$$a_{i+1} \preceq a[u_{i+1}u_i] \oplus a_i \preceq a[u_{i+1} \cdots u_1u_0],$$

concluding the induction step.

For $i = n-1$, we obtain $a_{n-1} \preceq a[u_{n-1} \cdots u_1u_0]$ with $a_{n-1} \in E(u_{n-1}, t)$, which is what we wanted to prove. □

Theorem B.4. *If isotonicity holds and all circuits are strictly inflationary, then the attributes elected at a node in stable state to reach a destination are the dominant attributes from the node to the destination, that is, $E(u, t) = A^*(u, t)$ in stable state for every nodes u and t.*

Proof. We show that $E(u, t) \subset A^*(u, t)$ and that $A^*(u, t) \subset E(u, t)$. Let $e \in E(u, t)$. From Lemma B.2, we known that e is the attribute of a simple path from u to t. Hence, from Lemma B.1, there is $a \in A^*(u, t)$ and a simple path from u to t with attribute a such that $a \preceq e$. Last, from Lemma B.3, there is $e' \in E(u, t)$ such that $e' \preceq a \preceq e$. Since the attributes of $E(u, t)$ are pairwise incomparable, it must be the case that $e' = a = e$. Thus, $e \in A^*(u, t)$ and $E(u, t) \subset A^*(u, t)$.

The argument for $A^*(u, t) \subset E(u, t)$ is analogous. Let $a \in A^*(u, t)$. There is path from u to t with attribute a. From Lemma B.3, there is $e \in E(u, t)$ such that $e \preceq a$. From Lemma B.2, e is the attribute of a simple path from u to t. Last, from Lemma B.1, there is $a' \in A^*(u, t)$ such that $a' \preceq e \preceq a$. The attributes of $A^*(u, t)$ are pairwise incomparable, so that $a' = e = a$. Thus, $a \in E(u, t)$ and $A^*(u, t) \subset E(u, t)$. □