Theory and New Primitives for Interconnecting Routing Protocol Instances

Franck Le[†] **Geoffrey G. Xie**[*] **Hui Zhang**[‡]

May 2009
CMU-CS-09-132

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

[†]Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA.
[*]Computer Science, Naval Postgraduate School, Monterey, CA, USA.
[‡]Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA.

This research was sponsored by the NSF under the 100x100 Clean Slate Project [1] (NSF-0331653), the 4D Project [2] (NSF-0520187), a Graduate Research Fellowship, and grants CNS-0520210, CNS-0721574. Views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of NSF, or the U.S. government.

Keywords: routing correctness, multi-instances, route selection, route redistribution

Abstract

Recent studies have shown that the primitives, that govern the interactions between routing protocol instances, are pervasively deployed in enterprise networks and the Internet but are extremely vulnerable to routing anomalies, including route oscillations and forwarding loops. In this paper, we propose a general theory for reasoning about routing properties across multiple routing instances. The theory is directly applicable to both link-state and vector routing protocols. Each routing protocol still makes independent routing decisions and may consider a variety of routing metrics, such as bandwidth, delay, cost, and reliability. While the theory permits a range of solutions, we focus on a design that requires no changes to existing routing protocols such as OSPF, RIP, EIRGP, and BGP. Guided by the theory, we derive a new set of primitives which are safe and more expressive than the current version, i.e., they can support operational goals not achievable today. We provide a detailed description of the new primitives in the form of IOS configuration commands and implemented them in the XORP routing software.

1 Introduction

Recent empirical studies [32, 28, 5] challenge the traditional, simple "BGP over your favorite IGP" view of the Internet routing architecture. They reveal that, in reality, the Internet routing landscape is much more complex as illustrated in Figure 1. Some ISPs, and many enterprise networks, deploy tens to hundreds of routing protocol instances simultaneously [28, 5], and the routing instances may be interconnected in diverse ways [32] (e.g., 57% of the analyzed networks in [5] have more than three routing instances – which is more than a single IGP and EGP –, and that study found both enterprise and university networks with more than ten instances. Other studies [28, 32] have also confirmed the prevalance of routing instances and exposed networks with even more than 400 routing instances.) The reasons behind those routing designs are various: They may derive from the need to route traffic based on different metrics, the desire for autonomy between departments of a same company [28], the requirement to filter route announcements [7], scalabity [28] or economical [5] reasons.

In this compound setting, researchers have brought to light the fundamental role of a set of primitives [28] that run on the border routers (e.g., A and B in Figure 1) and govern the interactions between different routing protocol instances. Even in the simplest "BGP over IGP" scenarios, those primtives are required to inject IGP or static routes into BGP. More importantly, operators use them to achieve critical design objectives (e.g., domain backup, shortest path routing across instances) that are infeasible using routing protocols (BGP, OSPF, IS-IS) alone [28].

Currently, the primitives responsible for the interconnections between routing instances consist of the so-called route selection and route redistribution procedures [10, 8]. Consider routers A and B in Figure 1. They are border routers in the sense that they belong to multiple routing protocol instances at the same time. For example, router A belongs to three routing protocol instances: BGP, OSPF 100, and RIP, and runs a separate routing process for each. As another example, router B is a member of two different OSPF instances. When a border router (e.g., A) receives routes from multiple routing processes (e.g., BGP, OSPF 100, RIP) for the same destination, the border router cannot directly compare the routes as each routing instance typically has its own metrics: e.g., RIP relies on a hop-count, whereas OSPF routes have a type (intra-area, inter-area, external type 1, external type 2) and a cost. The border router uses the route selection procedure to rank routes received from different routing processes and to determine which one to install in its forwarding table. The *route redistribution* procedure is required for interconnecting routing instances because, by default, routing processes of different protocol instances do not exchange routing information, even though they are on the same border router. Route redistribution must be explicitly enabled through router configuration. For example, the OSPF 200 and OSPF 300 instances will not exchange routing information unless mutual route redistribution between the two instances are configured on router B. Current operational networks rely heavily on these two procedures: e.g., a recent study [28] analyzed the usage of route redistribution in more than 1600 networks, and revealed that 99.9% of them depend on it.

It has been shown that the current mechanisms are extremely prone to misconfigurations [27, 29] and such errors are likely the root causes for many reported forwarding loops, route oscillations, prefix hijacks, and non-deterministic paths [30]. In response, several analytical models [27, 29, 30] have been developed enabling rigorous analyses of the current route selection and

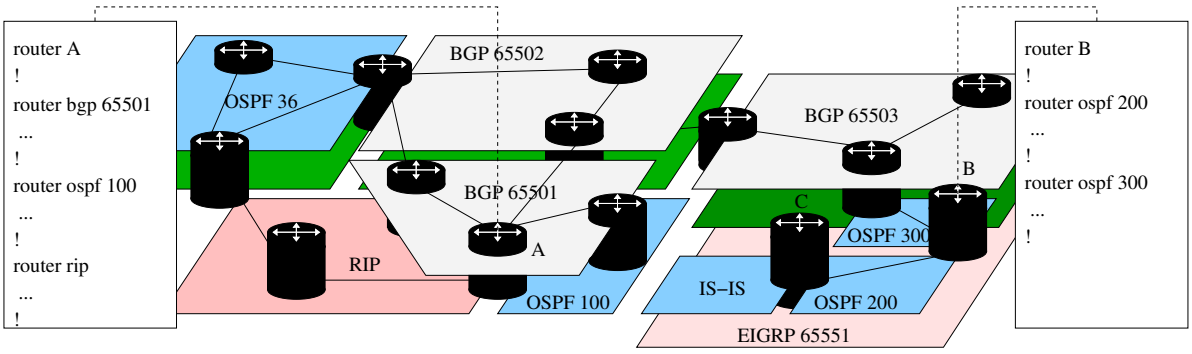

Figure 1: A typical slice of Internet routing landscape. Though abstracted, it still shows formidable complexity.

route redistribution procedures, and the formulation of practical configuration guidelines. However, adding band-aids to current mechanisms presents severe limitations: Configuration guidelines introduce new restrictions on setting parameters and, therefore, reduce the flexibility of the primitives and their power to implement operational goals. Operators have reported that the current primitives, even without any restriction, are already too rigid to support some desirable routing policies [28]. The existing analytical models are too tied to the current mechanisms and as such, do not provide insights for new designs.

We believe that the Internet will remain a myriad of routing protocol instances and that the primitives, responsible for the interconnections between routing protocol instances, will continue to play a crucial role in the Internet routing architecture. One single routing instance is unlikely to satisfy all operational requirements. Instead, driving forces behind the prevalence of routing protocol instances (e.g., the distinction between IGP and EGP functionality, the requirement to route traffic based on different metrics – bandwidth, network delay, hop count, path cost, load, reliability, etc. –, the desire for autonomy between departments) are likely to stay. In fact, the number of routing protocol instances may even grow with the emergence of new technologies (e.g., wireless networks, ad-hoc networks, vehicular ad-hoc networks, sensor networks, etc.) as each of them present unique characteristics and require distinct routing protocols. In this context, operators need a safe way to interconnect diverse routing instances.

This situation brings up a fundamental open question: Can we design a set of primitives – to interconnect routing protocol instances –, that both guarantee routing correctness – i.e., always converge to loop-free states regardless how they are configured – and increase the offered degree of expressiveness – allowing operators to fulfill their requirements? To answer the question, we need a theory for reasoning about routing across multiple routing protocol instances.

In this paper, we present such a theory for reasoning about routing correctness in networks with multiple interconnected routing instances. From the theory, we then derive a new set of primitives to interconnect routing protocol instances. While the theory permits a range of design options, we focus on a design characterized by no changes to existing routing protocols. We present the design details, including possible configuration commands, and explain why the new design is both safe and more expressive than the current design.

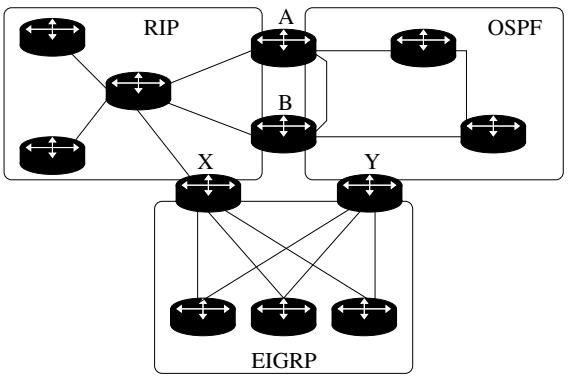

Figure 2: Incremental adoption of new primitives.

Our contributions are three-fold:

1. We have developed, to the best of our knowledge, the first formal framework for reasoning about routing properties in networks with *multiple* interconnected routing protocol instances. By introducing the novel concept of conversion functions, we are able to abstract the functional requirements of the interconnections between routing protocol instances. The key result is a set of sufficient conditions for guaranteeing correct routing, and optionally optimal paths, across multiple routing instances. The conditions allow operators to safely interconnect any combination of existing or future vectoring and link-state routing protocol instances. In addition, the conditions apply to routing protocols that consider not only one attribute (e.g., RIP, EIGRP) but also that perform a lexicographic comparison of multiple attributes (a_1, a_2, ..., a_k) to select the best route: $(a_1, a_2, \ldots, a_k) \leq (b_1, b_2, \ldots, b_k)$ if and only if $a_1 < b_1$ or ($a_1 = b_1$ and $a_2 < b_2$) and so on. For example, OSPF performs a lexicographic comparison of two attributes to determine the best route: the route type, and the route cost.

2. Guided by the new theory, we have created a set of new primitives to interconnect routing protocol instances. In contrast to the current mechanisms, our design makes it possible to guarantee routing safety regardless of configuration errors, while at the same time, supporting new operational goals not achievable today. Although the new primitives require routers' upgrade, the newly proposed primitives do not necessitate any modification to existing routing protocols (e.g., OSPF Link-State Advertisements, RIP Request/Reply messages, etc.). These properties enable an incremental adoption of the proposed solution and require only a partial deployment to fully take advantage of the offered features. For example, assuming the scenario depicted in Figure 2, upgrades to only the border routers (e.g., A, B, X, Y) would suffice to safely interconnect the three routing instances and implement new routing policies across them. We provide a detailed specification of all relevant procedures, configuration commands, and parameters.

3. We have analyzed the expressiveness of the proposed primitives with respect to several design goals considered important by the operational community. The results are compelling. While some of the goals are not feasible today, the new primitives are able to support all of the goals without requiring elaborate configurations.

4. Finally, we have implemented the new proposed primitives into the XORP (eXtensible Open source Routing Platform) [3] routing software. The pararemeters for the conversion functions can be specified either into a static configuration file – which is then loaded at XORP startup time – or through the XORP command line interface (CLI). The interactions between routing protocol instances are performed according to the newly defined route selection and route redistribution procedures. The implementation currently handles the interactions between BGP, OSPF and RIP instances. EIGRP being Cisco proprietary is not supported by XORP, and IS-IS is not yet supported as of XORP version 1.6. However, the theory and specifications also apply to EIGRP, IS-IS and later routing protocols.

The rest of the paper is structured as follows. Section 2 provides a brief description of the existing route selection and route redistribution procedures. Section 3 presents the newly proposed theory to reason about routing across multiple routing protocol instances. Section 4 identifies sufficient conditions to guarantee correct routing and optimal paths across routing protocol instances. Section 5 extends the framework and the results to routing protocols that lexicographically compares multiple attributes to select the best route. From the theory, Section 6 derives new primitives. Section 7 addresses the expressiveness of the new primitives. Section8 provides an overview of our implementation of the new primitives, and Section 9 discusses future work.

2 Background: Overview of Current Primitives

This section describes the two primitives, route selection and route redistribution, that currently govern the interactions between routing protocol instances.

Route selection: A router that runs multiple instances of different routing protocols (EIGRP, BGP, OSPF, etc.) or multiple instances of a same routing protocol (e.g., OSPF 100, OSPF 200, etc.) creates a separate routing process for each of them. In the rest of this paper, we will more formally say that two routing processes belong to the same routing protocol instance when the two processes are each on a different router, run the same routing protocol and exchange routing information through it.

For a given destination prefix P, each routing process selects one best route, from both the received updates and the local information, using a protocol specific algorithm: E.g., RIP simply compares the hop count while BGP uses an elaborate path ranking procedure. Then, if more than one routing processes offer a route to P, the router must perform a *route selection* procedure to determine which one to install in the Forwarding Information Base (FIB). This decision is currently based on a configurable parameter called Administrative Distance (AD) [9], with the preference given to the route with the lowest AD value. By default, in Cisco routers, RIP processes have an AD of 120 whereas OSPF processes have an AD of 110. As such, unless the AD are overridden, when receiving both a RIP and an OSPF route to the same destination prefix, a router prefers and installs the OSPF route in its FIB.

```
1   interface ethernet 0
2    ip address 192.1.1.1 255.255.255.0
3   !
4   interface ethernet 1
5    ip address 192.1.2.1 255.255.255.0
6   !
7   router rip
8    network 192.1.1.0
9    distance 100
10  !
11  router ospf 100
12   network 192.1.2.0 255.255.255.0 area 0.0.0.0
13   default-metric 100
14   redistribute rip metric 200 metric-type 1 subnets
15  !
```

Figure 3: Excerpt of a router configuration file illustrating the current IOS commands for route selection and route redistribution.

Route redistribution: Route redistribution allows operators to exchange routing information across routing instances. One complication is that routing protocols use different types of routing metrics. For example, RIP uses a single metric (hop count) while EIGRP relies on a weighted sum of bandwidth, delay, reliability, and load. The current route redistribution procedure handles this incompatibility in a crude fashion. It resets the metric of a redistributed route to either a default or a value manually configured by the operator. In either case, the new metric value typically has no relation to the route's original metric value.

Configuration commands: Each router vendor has its own configuration language. We focus on the Cisco IOS commands for illustration purposes. The syntax may differ across router vendors but the functions remain similar. Currently, configuring route selection and route redistribution on Cisco routers mainly involves three IOS commands. Each command allows a number of variants and options. Figure 3 illustrates an example use of these commands. The router has two interfaces, and runs two routing processes: RIP on the first interface, and OSPF on the second one.

1. The `distance` command (line 9) allows operators to override the default administrative distance of a routing process. In the depicted example, the administrative distance of RIP is set to 100, which is lower than the default administrative distance value of OSPF (110). Consequently, when receiving routes to the same destination from both RIP and OSPF, the router will select the RIP route.

2. The `redistribute` command (line 14) inside the OSPF command block activates route redistribution from RIP into the OSPF process. When configuring BGP, one may also use the `network` command to activate redistribution from *any* source (e.g., static, RIP, etc.) into BGP. Route filters can be applied to a `redistribute` or BGP `network` command to restrict the redistribution to a specific subset of routes. The `redistribute` command has protocol-specific options. For example, in the depicted example, the `metric-type` command is specific to OSPF, which mandates the routes to come in as "external type 1". A route can be injected into OSPF as either an external type 1 or an external type 2 route. The two types differ in the way their

costs will be calculated as they propagate inside the OSPF routing instance. The cost of a type 1 route will be dynamic, with the costs of the internal links added to the metric value assigned at the time of redistribution. In contrast, the cost of a type 2 route remains fixed regardless how many internal links it contains. In addition, a type 1 route is always prefered to a type 2 route.

3. The `default-metric` command (line 13) allows operators to configure a new default metric value for all route redistributions to a routing process. In addition, the `metric` option (line 14) may be used to override this default metric value for redistribution from a particular source. In the example, routes from the RIP routing process are injected into the OSPF process with an initial OSPF cost of 200 instead of the default value of 100.

In summary, the AD parameter (in route selection) and the metrics of newly redistributed routes (for route redistribution) are mainly set to arbitrary values, independently of the route's original attributes. As a result, information related to the initial routes (e.g., relative preference) may be lost potentially leading to persistent forwarding loops, permanent route oscillations and other unacceptable outcomes [29, 30].

3 Theory for multi-instance routing

Although considerable body of research has been devoted to the correctness of routing protocols, most prior work concentrated on individual routing protocol instances (e.g., RIP, OSPF, BGP). In contrast, this section presents a general framework to study routing properties *across multiple routing protocol instances*.

The proposed theory models routing protocols as algebras. Such an abstraction leaves out the specifics of the different routing protocols and focuses on the important properties. It allows the results to be applicable to both existing and future routing protocols.

Algebraic structures have been proposed to solve various network routing problems [6, 16, 19, 17, 23]. Section 3.1 provides a brief overview of the most recent and relevant results. Then, Section 3.2 introduces the notion of *conversion functions* to model the interactions between routing algebras and extend the focused analysis to a network with multiple routing protocol instances.

3.1 Background: Routing algebras

Recent works [36, 37, 23] have proposed modeling routing protocols as routing algebras, and have focused on identifying key properties to guarantee correct routing. Routing algebras can be viewed as an abstraction and generalization of shortest path routing. As illustrated in Figure 4 [23], each route has a signature ($\sigma \in \Sigma$) to model its relative precedence, and the notion of link weights is generalized to policy labels. When a route with signature σ is extended over a link ("u - v" in this example), with policy label $\lambda \in L$, the route's new signature becomes $(\lambda \oplus \sigma) \in \Sigma$. In other words, a signature represents the set of a route's attributes, a label represents the set of routing policies when a route is propagated over a given link, and $\oplus$ symbolizes the application of the routing policies to a route.

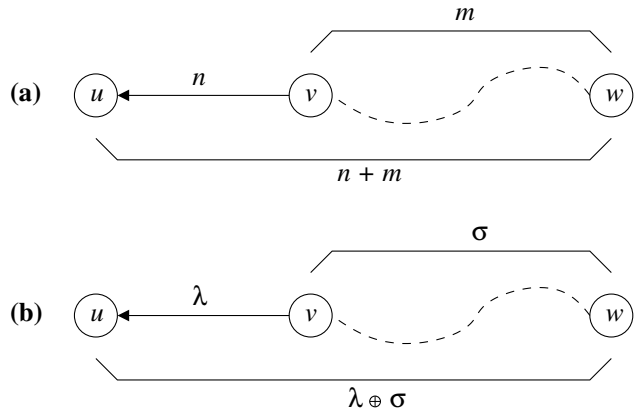

Figure 4: Illustration of similarity between (a) "classic" shortest path, and (b) routing algebra.

More formally, an algebra A is defined by a tuple (L, Σ, ϕ, $\oplus$, $\preceq$) [37] with ϕ being a special signature indicating a prohibited path. $\oplus$ is a mapping from $L \times \Sigma$ into Σ. The relation $\preceq$ is called a *preference relation* and creates a total pre-order over Σ. It allows to rank routes from A: If two routes have signatures α and β, (α, $\beta \in \Sigma$) and $\alpha \preceq \beta$, the route with signature α is prefered to the one with signature β. If $\alpha \preceq \beta$ and $\beta \preceq \alpha$, then we say that α and β are *equally prefered* (noted $\alpha \sim \beta$). Prohibited paths – paths with signature ϕ – are not further extended and $\forall \sigma \in \Sigma \setminus \{\phi\}, \sigma \prec \phi$.

The relation $\preceq$, being a total pre-order over Σ, satisfies the following properties:

(Reflexivity) $\forall \sigma \in \Sigma, \sigma \preceq \sigma$

(Transitivity) $\forall \sigma_1, \sigma_2, \sigma_3 \in \Sigma$, if $\sigma_1 \preceq \sigma_2$ and $\sigma_2 \preceq \sigma_3$, then $\sigma_1 \preceq \sigma_3$

(Totality) $\forall \sigma_1, \sigma_2 \in \Sigma$, $\sigma_1 \preceq \sigma_2$ or $\sigma_2 \preceq \sigma_1$

The relation $\preceq$ is not necessarily antisymmetric, i.e., for σ_1, σ_2 in Σ, $\sigma_1 \preceq \sigma_2$ and $\sigma_1 \preceq \sigma_2$ do not imply $\sigma_1 = \sigma_2$. This relaxation allows to enlargen the scope of covered routing protocols. In particular, the framework can include path vectoring routing protocols. To illustrate it, we assume that a signature σ consists of a sequence of identifiers (e.g., router identifiers or BGP Autonomous System Numbers), and $\sigma_1 \preceq \sigma_2$ if σ_1 has a shorter sequence of identifiers than σ_2: For example, (27, 36, 45) $\preceq$ (117, 234, 54, 810). We note that (10, 20, 30) $\preceq$ (50, 30, 80) and (50, 30, 80) $\preceq$ (10, 20, 30) but (10, 20, 30) $\neq$ (50, 30, 80). This operation is similar to the BGP AS PATH.

To serve as an example of a routing algebra, the RIP routing protocol can be modeled by the following one: L = $\{1, 2, \ldots, 16\}$, Σ = $\{1, 2, \ldots, 16\}$, ϕ = 16, "$\preceq$" = "$\leq$", and $\oplus$ defined as $\lambda \oplus \sigma = \min(\lambda + \sigma, \phi)$. Each hop in a path is assigned a configurable hop count which can take any value from 1 to 16. When a router receives a routing update, it adds its hop count to the metric value, and routes with hop count of 16 or more are prohibited and not propagated. In this specific case, $L = \Sigma$. However, this may not always be the case. As another example of routing algebras, a routing protocol that selects the path with maximum available bandwidth can be modeled with ($\oplus$, $\preceq$) = (min, max).

We consider m routing instances, and represent each of them by a distinct algebra $A_i = (L_i, \Sigma_i, \phi_i, \oplus_i, \preceq_i)$, $(1 \leq i \leq m)$. Previous work [36, 37] has identified sufficient conditions

	SM	I	$\oplus$ is associative
vectoring	✓		
link-state	✓	✓	✓

Table 1: Sufficient conditions for correctness for vectoring and link-state routing protocols.

for routing correctness for both vectoring and link state routing protocols[1]. These properties are summarized in Table 1 [23]. First, a routing algebra A_i satisfies the Strict monotonicity (SM) property if the following condition holds:

(SM) $\forall l \in L_i, \forall \sigma \in \Sigma_i \setminus \{\phi_i\}, \sigma \prec_i (l \oplus_i \sigma)$.

Strict monotonicity alone is a sufficient condition for routing correctness for a vectoring protocol: When a router further propagates a route, its preference must strictly decrease. For link-state protocols, additional properties are needed, and a routing algebra A_i satisfies the isotonicity (I) property[2] if the following conditions hold:

(Righ-Isotonicity) $\forall l \in L_i, \forall \sigma_1, \sigma_2 \in \Sigma_i$, if $\sigma_1 \preceq_i \sigma_2$, then $l \oplus_i \sigma_1 \preceq_i l \oplus_i \sigma_2$.

(Left-Isotonicity) $\forall \sigma_1, \sigma_2, \sigma_3 \in \Sigma_i$, if $\sigma_1 \preceq_i \sigma_2$, then $\sigma_1 \oplus_i \sigma_3 \preceq_i \sigma_2 \oplus_i \sigma_3$.

Isotonicity means that the preference order between two routes is preserved when they both are prepended by, or extended over, a common link. In fact, right-isotonicity (respectively, right and left isotonicity) is also a sufficient condition to guarantee optimal paths for vectoring (respectively, link-state) routing algebras [36, 18, 37].

Prior work used this elegant framework to analyze BGP and design new routing protocols through composition of routing algebras that are simple but conform to the sufficient conditions for correctness. However, the framework only applies to a network with a single routing protocol instance, i.e., every router in that network must run a single, identical routing protocol. The next section extends the framework to eliminate this limitation.

3.2 Conversion functions

We distinguish two types of routing algebras: *unary* algebras and *n-ary* algebras. Unary algebras use a single attribute to determine their best path. An example is RIP which selects the route with the lowest hop count. In contrast, we call n-ary algebras, algebras that perform a lexicographic comparison of up to n attributes. For example, BGP best path selection algorithm is a lexicographic order of the *local-preference*, the *AS-PATH length*, the *origin type*, and other additional attributes.

[1]By link-state, we refer to routing protocols where the routing information is flooded to all members, and each participant performs Dijkstra's algorithm to determine the selected paths. Examples of link-state routing protocols include OSPF and IS-IS.

[2]We adopt the terminology proposed in [36, 37]. However, other works have also called *monotonicity* the property herein named *isotonicity*, and used the denomination *nondecreasing* for the herein *monotonicity* property.

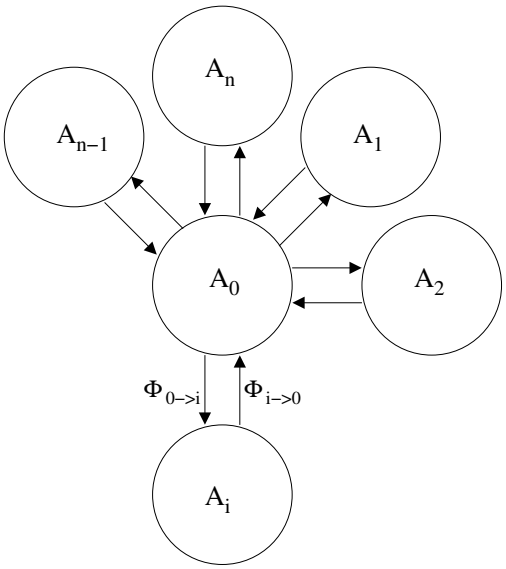

Figure 5: Illustration of the connections between routing algebras A_1, A_2, ..., A_m and the common algebra A_0.

We first define conversion functions for unary algebras. We will extend the results to the more general case of n-ary algebras in Section 5.

We observe that the heart of route selection and route redistribution procedures are two types of routing metric conversions. For route selection, metric conversions are required to establish a common ground to compare routes from different routing processes. For route redistribution, metric conversions are effectively performed when assigning metric values to redistributed routes within their new routing process.

Therefore, we propose to model the interactions between routing protocol instances as indirect connections between their respective routing algebras via a common algebra with a universal metric (signature) space. Moreover, we model the connection between each algebra A_i and the common algebra with a pair of conversion functions $\Phi_{i\rightarrow 0}()$ and $\Phi_{0\rightarrow i}()$ (Figure 5).

To illustrate the utility of these conversion functions in *comparing routes from different algebras*, let us assume a router $\mathcal{R}$ receiving routes to the same destination through k distinct routing algebras (A_1, A_2, ... A_k), and with signatures σ_1, σ_2, ..., and σ_k, respectively (See Figure 6.) Because these signatures belong to different signature domains, they can not be directly compared. Consequently, all σ_1, σ_2, ..., σ_k are first converted, through the conversion functions $\Phi_{1\rightarrow 0}()$, $\Phi_{2\rightarrow 0}()$,..., $\Phi_{k\rightarrow 0}()$, into the common *universal metric* space. Then, with the signatures being in the same unit, a best route can be selected.

Now, to exemplify the use of conversion functions in the *exchange of routing information across routing algebras*, let us assume that the router $\mathcal{R}$ is redistributing a route to P from A_i, with signature σ_i, into A_j. Such route redistribution allows routers in A_j to learn a route to P. For $\mathcal{R}$ to advertise the route in A_j, the initial signature, σ_i is first converted into universal metric, $\Phi_{i\rightarrow 0}(\sigma_i)$, and then mapped into a signature belonging to the target routing algebra's signature domain, Σ_j, through the conversion function $\Phi_{0\rightarrow j}()$: $\Phi_{0\rightarrow j} \circ \Phi_{i\rightarrow 0}(\sigma_i)$.

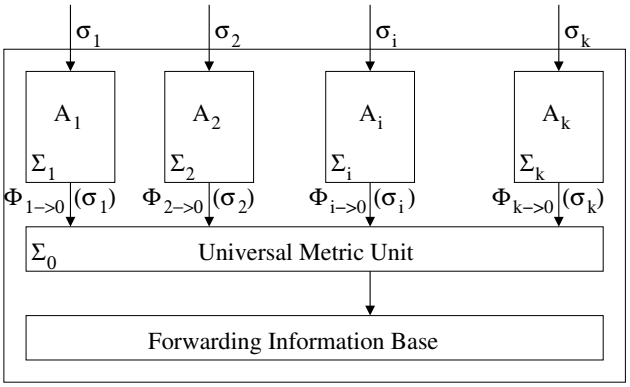

Figure 6: Ranking of routes received from different algebras.

More formally, let $L = L_1 \cup L_2 \cup ... \cup L_m$, $\Sigma = \Sigma_1 \cup \Sigma_2 \cup ... \cup \Sigma_m$, and A_0 $(\Sigma_0, \phi_0, \oplus_0, \preceq_0)$ denote the common ground algebra: Σ_0 represents the domain of signatures in universal metric units, and the relation $\preceq_0$ is a total pre-order over Σ_0.

Definition 1: Each algebra A_i $(i \in [1, m])$ is associated with two conversion functions:
1) $\Phi_{i \to 0}$: $\Sigma_i \to \Sigma_0$
2) $\Phi_{0 \to i}$: $\Sigma_0 \to \Sigma_i$

Definition 2: The binary relation $\preceq$ over Σ is defined as:
$\forall \alpha \in \Sigma_i, \beta \in \Sigma_j$,

$$\alpha \preceq \beta \stackrel{def}{=} \begin{cases} \alpha \preceq_i \beta & \text{if } i = j \\ \Phi_{i \to 0}(\alpha) \preceq_0 \Phi_{j \to 0}(\beta) & \text{else} \end{cases}$$

The relation $\preceq$ allows routers to rank any set of routes. If two routes are from the same routing algebra (A_i), the routing protocol specific best path selection algorithm $(\preceq_i)$ determines the best route. If routes are from different routing algebras, the signatures are first converted into universal metric units through the respective conversion functions, and the total pre-order $\preceq_0$ over Σ_0 defines the ranking.

Definition 3: The operator $\oplus : L \times \Sigma \to \Sigma$ is defined as
$\forall \lambda \in L_j, \sigma \in \Sigma_i$,

$$\lambda \oplus \sigma \stackrel{def}{=} \begin{cases} \lambda \oplus_j \sigma & \text{if } i = j \\ \lambda \oplus_j \Phi_{0 \to j} \circ \Phi_{i \to 0}(\sigma) & \text{else} \end{cases}$$

The operator $\oplus$ specifies the signature of a route as it is further propagated. In particular, for a route with a signature $\sigma, \sigma \in \Sigma_i$, to be extended over an arc with a label $\lambda, \lambda \in L_j$ $(i \neq j)$, the signature must first be converted into a signature of Σ_j, i.e., $\Phi_{0 \to j} \circ \Phi_{i \to 0}(\sigma)$, and the signature of the redistributed route then becomes $\lambda \oplus_j \Phi_{0 \to j} \circ \Phi_{i \to 0}(\sigma)$.

In summary, we have created a general formal framework to model the interactions between routing protocol instances. For example, consider today's route selection and route redistribution

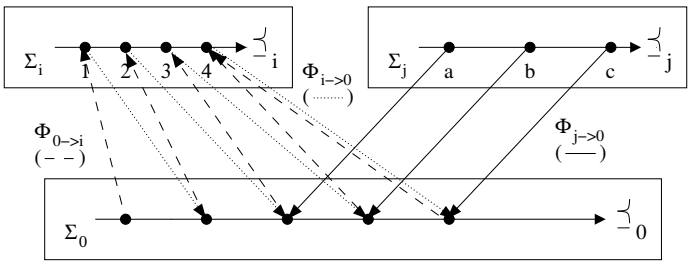

Figure 7: Illustration of Condition 1.

mechanisms. The current route selection relies on the notion of administrative distances (AD): each routing process is assigned a default AD value, which can be overridden by network operators to any arbitrary value. A route inherits the AD value of its routing process. When receiving routes to the same destination from multiple routing protocol instances, a router selects the route with the lowest AD value. AD values can range between 1 and 255, and a route with an AD value of 255 is considered invalid. As for route redistribution, when a route is injected into a new routing instance, its metric value is typically reset to a constant. Therefore, the current design of primitives can be modeled by constant conversion functions. The universal metric space is $\Sigma_0 = \{1, 2, \ldots, 255\}$, totally ordered by $\leq$ and with the prohibited signature $\phi_0 = 255$[3].

4 Safety conditions

The previous section introduced the notion of conversion functions to model and reason about the interconnections between routing protocol instances. Using this framework, the initial question on whether we can redesign safer and more expressive primitives can be more specifically reformulated as: What conditions must the conversion functions satisfy?

This section presents sufficient conditions for the conversion functions to guarantee correct routing, and optimal paths, across multiple routing instances.

Condition 1: $\forall i \in [1, m]$,
(a) $\Phi_{i\to 0} : \Sigma_i \to \Sigma_0$ is strictly increasing, i.e., $\forall \sigma_1, \sigma_2 \in \Sigma_i, \sigma_1 \prec_i \sigma_2 \Rightarrow \Phi_{i\to 0}(\sigma_1) \prec_0 \Phi_{i\to 0}(\sigma_2)$
(b) $\forall \sigma \in \Sigma_0$, $\sigma \preceq_i \Phi_{i\to 0} \circ \Phi_{0\to i}(\sigma)$

This condition stipulates that the conversion function $\Phi_{i\to 0}$ maps distinct signatures of Σ_i into distinct values of Σ_0 in an order preserving manner (Condition 1a). In addition, the preference of a route should not decrease as it is redistributed (Condition 1b). Figure 7 illustrates the above conditions: the pair of conversion functions ($\Phi_{i\to 0}$, $\Phi_{0\to i}$) satisfies Condition 1.

Lemma 1: *Condition 1 guarantees that the relation $\preceq$ is a total pre-order over the set of signatures Σ.*

[3]Certain router vendor options allow the configuration of AD per prefix. However, the framework aims at representing the interactions in general.

Proof: To prove that the relation $\preceq$ is a total pre-order over Σ , we demonstrate that $\preceq$ is reflexive, transitive, and total.

• *Reflexivity:* $\forall a \in \Sigma, a \preceq a$.

We assume a in Σ. Because $\Sigma = \Sigma_1 \cup \Sigma_2 \cup ... \cup \Sigma_m$, there exists $i \in [1, m]$ such that $a \in \Sigma_i$. Then, since by definition $\preceq_i$ is a total pre-order over Σ_i, $\preceq_i$ is reflexive. In particular, $a \preceq_i a$. Finally, by definition of $\preceq$, we conclude that $a \preceq a$.

• *Transitivity:* $\forall a, b, c \in \Sigma$, $a \preceq b, b \preceq c \Rightarrow a \preceq c$.

We assume $a, b, c \in \Sigma$ such that $a \preceq b$ and $b \preceq c$. Since $\Sigma = \Sigma_1 \cup \Sigma_2 \cup ... \cup \Sigma_m$, $\exists i, j, k \in [1, m]$ such that $a \in \Sigma_i$, $b \in \Sigma_j$ and $c \in \Sigma_k$.

Case 1: $i = j = k$.

$a \preceq b \Rightarrow a \preceq_i b$ (Definition 2)

$b \preceq c \Rightarrow b \preceq_i c$ (Definition 2)

Then, as $\preceq_i$ is a total pre-order over Σ_i and transitive, we derive $a \preceq_i c$ and conclude that $a \preceq c$ (Definition 2).

Case 2: i, j, k all distinct. We have

$a \preceq b \Rightarrow \Phi_{i\to 0}(a) \preceq_0 \Phi_{j\to 0}(b)$ (Definition 2) and

$b \preceq c \Rightarrow \Phi_{j\to 0}(b) \preceq_0 \Phi_{k\to 0}(c)$ (Definition 2)

Then, as $\preceq_0$ is a total pre-order over Σ_0, we derive $\Phi_{i\to 0}(a) \preceq_0 \Phi_{k\to 0}(c)$, which implies that $a \preceq c$.

Case 3: $i = j \neq k$. We have

$a \preceq b \Rightarrow a \preceq_i b$ (Definition 2)

$\Rightarrow \Phi_{i\to 0}(a) \preceq_0 \Phi_{j\to 0}(b)$ (Condition 1(a)) and

$b \preceq c \Rightarrow \Phi_{j\to 0}(b) \preceq_0 \Phi_{k\to 0}(c)$ (Definition 2)

Then, as $\preceq_0$ is a total pre-order over Σ_0, we derive $\Phi_{i\to 0}(a) \preceq_0 \Phi_{k\to 0}(c)$, which implies that $a \preceq c$.

Case 4: $i \neq j = k$. We have

$a \preceq b \Rightarrow \Phi_{i\to 0}(a) \preceq_0 \Phi_{j\to 0}(b)$ (Definition 2)

$\Rightarrow \Phi_{i\to 0}(a) \preceq_0 \Phi_{j\to 0}(b)$ (Condition 1(a)) and

$b \preceq c \Rightarrow b \preceq_j c$ (Definition 2)

$\Rightarrow \Phi_{j\to 0}(b) \preceq_0 \Phi_{j\to 0}(c)$ (Condition 1(a))

Then, as $\preceq_0$ is a total pre-order over Σ_0, we derive $\Phi_{i\to 0}(a) \preceq_0 \Phi_{j\to 0}(c)$, which implies that $a \preceq c$.

Case 5: $i = k \neq j$. We have

$a \preceq b \Rightarrow \Phi_{i\to 0}(a) \preceq_0 \Phi_{j\to 0}(b)$ (Definition 2)

$b \preceq c \Rightarrow \Phi_{j\to 0}(b) \preceq_0 \Phi_{i\to 0}(c)$ (Definition 2)

Then, as $\preceq_0$ is a total pre-order over Σ_0, we derive $\Phi_{i\to 0}(a) \preceq_0 \Phi_{i\to 0}(c)$.

This inequation implies that $a \preceq_i c$. Otherwise, $a \succ_i c$. Then, from Condition 1(a), we obtain: $\Phi_{i\to 0}(a) \succ_0 \Phi_{i\to 0}(c)$. This contradicts the previous inequation.

To conclude, we have $a \succ_i c$, which implies that $a \preceq c$.

• *Totality:* $\forall a, b \in \Sigma$, $a \preceq b$ or $b \preceq a$. We assume $a, b \in \Sigma$. Since $\Sigma = \Sigma_1 \cup \Sigma_2 \cup ... \cup \Sigma_m$, $\exists i, j \in [1, m]$ such that $a \in \Sigma_i$ and $b \in \Sigma_j$.

Case 1: $i = j$. We have $a, b \in \Sigma_i$, and since $\preceq_i$ is a total pre-order over Σ_i, we have either $a \preceq_i b$ or $b \preceq_i a$. By definition of $\preceq$ (Definition 2), we then derive that $a \preceq b$ or $b \preceq a$.

Case 2: $i \neq j$. Since $\preceq_0$ is a total pre-order over Σ_0, we have either $\Phi_{i\to 0}(a) \preceq_0 \Phi_{j\to 0}(b)$ or $\Phi_{j\to 0}(b) \preceq_0 \Phi_{i\to 0}(a)$. Then, by definition of $\preceq$ (Definition 2), we conclude that $a \preceq b$ or $b \preceq a$. □

Theorem 1: *If all algebras $A_1, A_2, ..., A_m$ are SM, then Condition 1 is a sufficient condition to guarantee the preservation of the SM property within and across the algebras, i.e., $\forall \lambda \in L$, $\forall \sigma \in \Sigma$, $\sigma \prec (\lambda \oplus \sigma)$.*

Proof: To demonstrate that Condition 1 is a sufficient condition to preserve SM across the algebras, we assume a router receiving a route with signature σ from algebra A_i, i.e., $\sigma \in \Sigma_i$, and we assume that the route is extended to an arc with a label λ. We show that assuming that all algebras $A_1, A_2, ..., A_m$ are SM, and that the conversion functions are compliant with Condition 1, then the extended route has a strictly lower preference than the initial route. There are two cases:

Case 1: $\lambda \in L_i$. The initial route is extended into the same routing algebra A_i. Then, since A_i is SM, we conclude that $\sigma \prec_i \lambda \oplus_i \sigma$, i.e., $\sigma \prec \lambda \oplus \sigma$ (according to Definitions 2 and 3).

Case 2: $\lambda \in L_j, j \neq i$. The initial route from A_i is extended into a different algebra A_j.

Since A_j is SM,

$\Phi_{0\to j} \circ \Phi_{i\to 0}(\sigma) \prec_j \lambda \oplus_j \Phi_{0\to j} \circ \Phi_{i\to 0}(\sigma)$

Then, by Condition 1(a),

$\Phi_{j\to 0} \circ \Phi_{0\to j} \circ \Phi_{i\to 0}(\sigma) \prec_0 \Phi_{j\to 0}(\lambda \oplus_j \Phi_{0\to j} \circ \Phi_{i\to 0}(\sigma))$

From Condition 1(b), we also have

$\Phi_{i\to 0}(\sigma) \preceq_0 \Phi_{j\to 0} \circ \Phi_{0\to j} \circ \Phi_{i\to 0}(\sigma)$

Given that $\preceq_0$ is transitive, we obtain from the two above inequations

$\Phi_{i\to 0}(\sigma) \prec_0 \Phi_{j\to 0}(\lambda \oplus_j \Phi_{0\to j} \circ \Phi_{i\to 0}(\sigma))$

By definition of $\oplus$, we get

$\Phi_{i\to 0}(\sigma) \prec_0 \Phi_{j\to 0}(\lambda \oplus \sigma)$

Finally, by definition of $\preceq$, we conclude

$\sigma \prec \lambda \oplus \sigma$ □

Theorem 1 is an important result. When the exchange of routing information across routing algebras is performed in a vectoring fashion – i.e., when the re-advertisement of a route into a new routing algebra consists of (1) a destination, (2) some metrics, and (3) a direction towards the destination – then from the properties described in Table 1, we derive that to guarantee correct routing, it suffices for the algebras to be SM and for the conversion functions to comply to Condition 1.

We note that the current route selection and route redistribution primitives (Section 2), which can be modeled by constant conversion functions (Section 3), satisfy neither Condition 1(a) nor Condition 1(b).

Condition 2: $\forall i \in [1, m], \Phi_{0\to i} : \Sigma_0 \to \Sigma_i$ is increasing, i.e.,
$\forall \sigma_1, \sigma_2 \in \Sigma_0, \sigma_1 \preceq_0 \sigma_2 \Rightarrow \Phi_{0\to i}(\sigma_1) \preceq_i \Phi_{0\to i}(\sigma_2)$

Theorem 2: *If all algebras $A_1, A_2, ..., A_m$ are right-isotone, then Conditions 1 and 2 guarantee the preservation of the right-isotonicity property within and across the algebras, i.e.,*
$\forall l \in L, \forall \sigma_1, \sigma_2 \in \Sigma$, *if* $\sigma_1 \preceq \sigma_2$, *then* $l \oplus \sigma_1 \preceq l \oplus \sigma_2$.

Proof: We assume $l \in L$, and $\sigma_1, \sigma_2 \in \Sigma$ with $\sigma_1 \preceq \sigma_2$. Since $L = L_1 \cup L_2 \cup ... \cup L_m$, and $\Sigma = \Sigma_1 \cup \Sigma_2 \cup ... \cup \Sigma_m$, $\exists i, j, k \in [1, m]$ such that $l \in L_i$, $\sigma_1 \in \Sigma_j$ and $\sigma_2 \in \Sigma_k$. We distinguish five cases:

1) $i = j = k$,
2) i, j, k all distinct,
3) $i = j \neq k$,
4) $i \neq j = k$,
5) $i = k \neq j$.

For the first case, the proof simply derives from the fact that A_i is right-isotone. We now address the second case. For the three other cases (3, 4, 5), the reasoning is similar. We assume that i, j and k are all distinct.

We show that $\sigma_1 \preceq \sigma_2 \Rightarrow l \oplus \sigma_1 \preceq l \oplus \sigma_2$.
$\sigma_1 \preceq \sigma_2$
$\Rightarrow \Phi_{j\to 0}(\sigma_1) \preceq_0 \Phi_{k\to 0}(\sigma_2)$
(by definition of $\preceq$))

$\Rightarrow \Phi_{0\to i} \circ \Phi_{j\to 0}(\sigma_1) \preceq_i \Phi_{0\to i} \circ \Phi_{k\to 0}(\sigma_2)$
(because of Condition 2(a))
$\Rightarrow l \oplus_i \Phi_{0\to i} \circ \Phi_{j\to 0}(\sigma_1) \preceq_i l \oplus_i \Phi_{0\to i} \circ \Phi_{k\to 0}(\sigma_2)$
(by right-isotonicity of $\oplus_i$)
$\Rightarrow l \oplus \sigma_1 \preceq_i l \oplus \sigma_2$
(by definition of $\oplus$)
$\Rightarrow l \oplus \sigma_1 \preceq l \oplus \sigma_2$
(by definition of $\preceq$) □

Theorem 2 guarantees optimal paths when the exchange of routing information across routing algebras is perfomed in a vectoring manner.

Condition 3:
(a) $\forall i \in [1, m]$, $\Phi_{i\to 0}$ is bijective and $\Phi_{0\to i} = \Phi^{-1}_{i\to 0}$
(b) $\forall i \in [1, m]$, $\Phi_{i\to 0}$ is distributive, i.e., $\forall \sigma_1, \sigma_2 \in \Sigma_i$, $\Phi_{i\to 0}(\sigma_1 \oplus_i \sigma_2) = \Phi_{i\to 0}(\sigma_1) \oplus_0 \Phi_{i\to 0}(\sigma_2)$
(c) $\forall i \in [1, m]$, $\Phi_{0\to i}$ is distributive, i.e., $\forall \sigma_1, \sigma_2 \in \Sigma_0$, $\Phi_{0\to i}(\sigma_1 \oplus_0 \sigma_2) = \Phi_{0\to i}(\sigma_1) \oplus_i \Phi_{0\to i}(\sigma_2)$

Theorem 3: *If all algebras A_0, A_1, ..., A_m are isotone, and for every i in [1,m], $\oplus_i$ is associative, then Conditions 1 to 3 guarantee the isotonicity property across the algrebas and the associativity of $\oplus$.*

Proof: We first address the preservation of the isotonicity property across the algebras. The right-isotonicity derives from Theorem 2. Therefore, we only need to prove that $\forall \sigma_1, \sigma_2, \sigma_3 \in \Sigma$, $\sigma_1 \preceq \sigma_2 \Rightarrow \sigma_1 \oplus \sigma_3 \preceq \sigma_2 \oplus \sigma_3$.

We assume $\sigma_1, \sigma_2, \sigma_3 \in \Sigma$. Since $\Sigma = \Sigma_1 \cup \Sigma_2 \cup ... \cup \Sigma_m$, $\exists i, j, k \in [1, m]$ such that $\sigma_1 \in \Sigma_i$, $\sigma_2 \in \Sigma_j$ and $\sigma_3 \in \Sigma_k$. We distinguish five cases:
1) $i = j = k$,
2) i, j, k all distinct,
3) $i = j \neq k$,
4) $i \neq j = k$,
5) $i = k \neq j$.
We focus on the case where i, j, k are all distinct. For the other cases, the reasoning is similar.
$\sigma_1 \preceq \sigma_2$
$\Rightarrow \Phi_{i\to 0}(\sigma_1) \preceq_0 \Phi_{j\to 0}(\sigma_2)$
(by definition of $\preceq$ or Condition 1(a))
$\Rightarrow \Phi_{i\to 0}(\sigma_1) \oplus_0 \Phi_{k\to 0}(\sigma_3) \preceq_0 \Phi_{j\to 0}(\sigma_2) \oplus_0 \Phi_{k\to 0}(\sigma_3)$
(because A_0 is left-isotone)
$\Rightarrow \Phi_{i\to 0}(\sigma_1 \oplus_i \Phi_{0\to i} \circ \Phi_{k\to 0}(\sigma_3)) \preceq_0 \Phi_{j\to 0}(\sigma_2 \oplus_j \Phi_{0\to j} \circ$
$\Phi_{k\to 0}(\sigma_3))$
(because $\Phi_{i\to 0}()$ and $\Phi_{j\to 0}()$ are bijective and distributive)
$\Rightarrow \Phi_{i\to 0}(\sigma_1 \oplus \sigma_3) \preceq_0 \Phi_{j\to 0}(\sigma_2 \oplus \sigma_3)$
(by definition of $\oplus$)

$\Rightarrow \sigma_1 \oplus \sigma_3 \preceq \sigma_2 \oplus \sigma_3$
(by definition of $\preceq$)

We now demonstrate the associativity of $\oplus$, i.e., $\forall\ \sigma_1, \sigma_2, \sigma_3 \in \Sigma$, $(\sigma_1 \oplus \sigma_2) \oplus \sigma_3 = \sigma_1 \oplus (\sigma_2 \oplus \sigma_3)$. We assume $\sigma_1, \sigma_2, \sigma_3 \in \Sigma$. Since $\Sigma = \Sigma_1 \cup \Sigma_2 \cup ... \cup \Sigma_m$, $\exists i, j, k \in [1, m]$ such that $\sigma_1 \in \Sigma_i$, $\sigma_2 \in \Sigma_j$ and $\sigma_3 \in \Sigma_k$. We again focus and illustrate only the case where i, j, k are all distinct since the other cases can be demonstrated through analogous steps. We expand the two expressions:

1. $(\sigma_1 \oplus \sigma_2) \oplus \sigma_3$
   = $(\sigma_1 \oplus_i \Phi_{0\to i} \circ \Phi_{j\to 0}(\sigma_2)) \oplus \sigma_3$
   (by definition of $\oplus$)
   = $(\sigma_1 \oplus_i \Phi_{0\to i} \circ \Phi_{j\to 0}(\sigma_2)) \oplus_i \Phi_{0\to i} \circ \Phi_{k\to 0}(\sigma_3)$
   (by definition of $\oplus$)
   = $\sigma_1 \oplus_i (\Phi_{0\to i} \circ \Phi_{j\to 0}(\sigma_2) \oplus_i \Phi_{0\to i} \circ \Phi_{k\to 0}(\sigma_3))$
   (by assoc. of $\oplus_i$)

2. $\sigma_1 \oplus (\sigma_2 \oplus \sigma_3)$
   = $\sigma_1 \oplus (\sigma_2 \oplus_j \Phi_{0\to j} \circ \Phi_{k\to 0}(\sigma_3))$
   (by definition of $\oplus$)
   = $\sigma_1 \oplus_i (\Phi_{0\to i} \circ \Phi_{j\to 0}(\sigma_2 \oplus_j \Phi_{0\to j} \circ \Phi_{k\to 0}(\sigma_3)))$
   (by definition of $\oplus$)
   = $\sigma_1 \oplus_i (\Phi_{0\to i} \circ \Phi_{j\to 0}(\sigma_2) \oplus_i \Phi_{0\to i} \circ \Phi_{j\to 0} \circ \Phi_{0\to j} \circ$
   $\Phi_{k\to 0}(\sigma_3)))$
   (because $\Phi_{0\to i}()$ and $\Phi_{j\to 0}()$ are distributive)
   = $\sigma_1 \oplus_i (\Phi_{0\to i} \circ \Phi_{j\to 0}(\sigma_2) \oplus_i \Phi_{0\to i} \circ \Phi_{k\to 0}(\sigma_3)))$
   (because of Condition 3(a))

From the above two results, we conclude
$(\sigma_1 \oplus \sigma_2) \oplus \sigma_3 = \sigma_1 \oplus (\sigma_2 \oplus \sigma_3)$ □

When exchanging routes across multiple routing protocols instances, the exchange can be performed either in a vectoring manner or in a link-state manner. For example, today's route redistribution exchanges routes in a vectoring manner since an advertisement primarily consists of a destination, a direction and some metric. The receiving router only knows the next-hop and cannot reconstruct a map of the network topology. In contrast, when the exchange of routing information is performed in a link-state manner, an advertisement would be similar to an OSPF Link-State Advertisement; and routers could learn the topologies of other routing instances. The theory supports both modes, and Theorem 3 guarantees routing correctness and optimal paths when the exchange of routing information across the algebras is performed in a link-state manner. This form of route redistribution enables routers in one link-state protocol instance to learn the complete topology of other neighboring instances. This information permit more sophisticated forms of routing such as

	Routing Correctness	Path Optimality
vectoring	Condition 1	Conditions 1, 2
link-state	Conditions 1, 2, 3	Conditions 1, 2, 3

Table 2: Sufficient conditions for the conversion functions to guarantee correct routing and optimal paths.

disjoint paths routing across different instances.

Table 2 summarizes the sufficient conditions for the conversion functions, in order to guarantee correct routing and optimal paths.

5 Generalization to n-ary algebras

The previous section addresses routing protocols that consist of a single metric (e.g., RIP, EIGRP). However, many of the existing routing protocols perform a lexicographic comparison of more than one attributes: For example, OSPF first looks at the route type (intra-area, inter-area, external) to determine the preference of the routes. Then, the OSPF cost is used as a tie-breaker between routes of the same type. Similarly, BGP sequentially examines multiple attributes to decide the best route.

As such, this section generalizes the definition of conversion functions and safety conditions to n-ary routing algebras. Each n-ary routing algebra A_i is defined as the lexicographical product of n unary routing algebras: $\otimes(A_{i1}, A_{i2}, \ldots, A_{in})$ [23]. Given two signatures in Σ_i, A_i first considers their first attribute from A_{i1} to determine the most preferred route. If the two routes have equal preference based on the first criteria, the second attribute, from A_{i2}, is considered and so forth. We note that unary algebras are a special case of n-ary algebras, where $n = 1$.

More formally, $\forall\, i \in [1, m]$, $A_i = \otimes(A_{i1}, A_{i2}, \ldots, A_{in})$ with $\forall\, i \in [1, m]$, $\forall\, d \in [1, n]$, A_{id} being an unary routing algebra (L_{id},Σ_{id},ϕ_{id},$\oplus_{id}$, $\preceq_{id}$), and the relation $\preceq_i$ over Σ_i defined as:
$\forall\, \alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n) \in \Sigma_i = (\Sigma_{i1}, \Sigma_{i2}, \ldots, \Sigma_{in})$,
$\forall\, \beta = (\beta_1, \beta_2, \ldots, \beta_n) \in \Sigma_i = (\Sigma_{i1}, \Sigma_{i2}, \ldots, \Sigma_{in})$,
$\alpha \preceq_i \beta \Leftrightarrow \exists e \in [1, n], \forall d < e, \alpha_d \sim_{id} \beta_d$ and ($\alpha_d \prec_{id} \beta_d$
or ($e = n$ and $\alpha_n \preceq_{in} \beta_n$)).

In this context of n-ary routing algebras, we extend the notion of conversion functions, the relation $\preceq$ and the operator $\oplus$ to compare and exchange routes between n-ary routing instances. We introduce a set of n unary algebras A_{01}, A_{02}, …, A_{0n} to compare routes from the different algebras. For $d \in [1, n]$, A_{0d} is represented by a tuple $(\Sigma_{0d}, \phi_{0d}, \oplus_{0d}, \preceq_{0d})$ with $\preceq_{0d}$ being a total pre-order over Σ_{0d}. Because of their similarities with those presented previously, proofs from this section are not detailed.

Definition 4: $\forall i \in [1, m], \forall d \in [1, n]$, A_{id} is associated with two conversion functions:
1) $\Phi_{id \to 0d}$: $\Sigma_{id} \to \Sigma_{0d}$, and
2) $\Phi_{0d \to id}$: $\Sigma_{0d} \to \Sigma_{id}$.

We note

1. $\forall i \in [1,m], L_i = (L_{i1}, L_{i2}, \ldots, L_{in})$,
2. $L = \cup_{1 \leq k \leq m} L_k$,
3. $\forall i \in [1,m], \Sigma_i = (\Sigma_{i1}, \Sigma_{i2}, \ldots, \Sigma_{in})$,
4. $\Sigma = \cup_{1 \leq k \leq m} \Sigma_k$

Definition 5: The binary relation $\preceq$ over Σ is defined as:
$\forall\ \alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n) \in \Sigma_i$,
$\forall\ \beta = (\beta_1, \beta_2, \ldots, \beta_n) \in \Sigma_j$,
$\alpha \preceq \beta \stackrel{def}{=}$
if $i = j$: $\alpha \preceq_i \beta$,
else: $\exists e \in [1,n], \forall d < e, \Phi_{id \to 0d}(a_d) \sim_{0d} \Phi_{jd \to 0d}(b_d)$ and
$(\Phi_{ie \to 0e}(a_e) \prec_{0e} \Phi_{je \to 0e}(b_e)$ or $(e = n$ and $\Phi_{in \to 0n}(a_n) \preceq_{0n} \Phi_{jn \to 0n}(b_n))$.

Definition 6: The operator $\oplus : L \times \Sigma \to \Sigma$ is defined as:
$\forall\ \lambda = (\lambda_1, \lambda_2, \ldots, \lambda_n) \in L_j$,
$\forall\ \sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n) \in \Sigma_i$,
$\lambda \oplus \sigma \stackrel{def}{=}$
if $i = j$: $(\lambda_1 \oplus_{i1} \sigma_1, \lambda_2 \oplus_{i2} \sigma_2, \ldots, \lambda_n \oplus_{in} \sigma_n)$
else: $(\lambda_1 \oplus_{i1} \Phi_{01 \to i1} \circ \Phi_{j1 \to 01}(\sigma_1)$,
$\lambda_2 \oplus_{i2} \Phi_{02 \to i2} \circ \Phi_{j2 \to 02}(\sigma_2)$,
$\ldots$,
$\lambda_n \oplus_{in} \Phi_{0n \to in} \circ \Phi_{jn \to 0n}(\sigma_n))$.

Condition 4: $\forall i \in [1,m], \forall d \in [1,n]$,
(a) $\Phi_{id \to 0d}$ is stricly increasing, i.e., $\forall \sigma_1, \sigma_2 \in \Sigma_{id}, \sigma_1 \prec_{id} \sigma_2 \Rightarrow \Phi_{id \to 0d}(\sigma_1) \prec_{0d} \Phi_{id \to 0d}(\sigma_2)$
(b) $\forall \sigma \in \Sigma_{0d}, \sigma \preceq_{0d} \Phi_{id \to 0d} \circ \Phi_{0d \to id}(\sigma)$

Lemma 2: *Condition 4 guarantees that the relation $\preceq$ is a total pre-order over the set of n-ary signatures Σ.*

Condition 5: $\forall i \in [1,m], \exists e \in [1,n]$ such that $\forall d < e$, A_{id} is either M or SM and A_{ie} is SM.

$$A_i = \otimes \quad (\underbrace{A_{i1}}_{\text{(M)}}, \underbrace{A_{i2}}_{\text{(M)}}, \underbrace{\ldots}_{\text{(M)}}, \underbrace{A_{ie}}_{\text{(SM)}}, \underbrace{\ldots}_{\text{Dont}}, \underbrace{A_{in}}_{\text{care}})$$

Theorem 4: *Conditions 4 and 5 guarantee the preservation of the SM property across the algebras, i.e.,* $\forall \lambda \in L, \forall \sigma \in \Sigma, \sigma \prec (\lambda \oplus \sigma)$.

Theorem 4 guarantees routing correctness when the exchange of routing information across the algebras is performed in a vectoring fashion. We observe that theorem 4 allows redistribution across algebras with different number of attributes: e.g., A_i uses m attributes while A_j uses $n \neq m$

attributes.

Condition 6: $\forall i \in [1,m], \forall d \in [1,n-1]$, $\Phi_{id \to 0d}$ are strictly increasing, and $\Phi_{in \to 0d}$ is increasing.

Theorem 5: *If for every i in [1,m], d in [1,n-1], A_{id} is stricly right-isotone, and A_{in} is right-isotone, then Conditions 4 and 6 guarantee right-isotonicity across the algebras.*

Theorem 5 guarantees optimal paths when the exchange of routing information across the algebras is performed in a vectoring fashion.

Condition 7: $\forall i \in [1,m], \forall d \in [1,n]$, $\Phi_{id \to 0d}$ are bijective and $\Phi_{id \to 0d}$ is the inverse function of $\Phi_{id \to 0d}$. In addition, $\Phi_{id \to 0d}$ and $\Phi_{0d \to id}$ are distributive, i.e.,
$\forall \sigma_1, \sigma_2 \in \Sigma_{id}$, $\Phi_{id \to 0d}(\sigma_1 \oplus_{id} \sigma_2) = \Phi_{id \to 0d}(\sigma_1) \oplus_{0d} \Phi_{id \to 0d}(\sigma_2)$
$\forall \sigma_1, \sigma_2 \in \Sigma_{0d}$, $\Phi_{0d \to id}(\sigma_1 \oplus_{0d} \sigma_2) = \Phi_{0d \to id}(\sigma_1) \oplus_{id} \Phi_{0d \to id}(\sigma_2)$

Theorem 5: *If (1) $\forall i \in [0,m], \forall d \in [1,n-1]$, A_{id} is strictly isotone, (2) A_{in} is isotone, and (3) $\forall i \in [1,m], d \in [1,n]$, $\oplus_{id}$ is associative, then Conditions 4, 6 and 7 guarantee the preservation of the isotonicity (I) property across the algebras and the associativity of $\oplus$.*

6 A new set of primitives

The theory presented in the previous two sections opens up new design possibilities, including the creation of a new class of primitives that inherently conforms to the identified safety conditions and hence guarantees routing correctness across routing protocol instances. This section describes the details of one such design.

Since many of the existing routing protocols are n-ary algebras (i.e., perform a lexicographic comparison of multiple attributes to determine the best route), we adopt the n-ary framework of Section 5 to reason about the interconnections between routing protocol instances.

We first describe some design decisions stemmed from our requirement not to modify existing routing protocols. We then present the universal metric space and default parameters for our design. Finally, we give a detailed specification of the new route selection and redistribution procedures and their configuration commands.

Design for incremental deployment: To enable incremental deployment of the new proposed primitives, we target a design that requires no modification to existing routing protocols (e.g., BGP, EIGRP, OSPF, RIP, IS-IS). This incremental deployment objective introduces two complications that constrain our design space.

The theory assumes each routing protocol instance to be correct and concentrates on conditions for the conversion functions. As such, the theory requires the routing instances to satisfy different conditions (e.g., Condition 5). However, we first observe that OSPF External Type 2 routes violate these requirements. In particular, OSPF External Type 2 routes violate the SM condition required

in Condition 5: As explained in Section 2, OSPF does not increase the cost of an External Type 2 route while the route propagates. We solve this problem by imposing the following restriction on the design:

OSPF-specific Restriction: *Routes redistributed into OSPF are always set to External Type 1.*

Second, prior work [23] has also shown that BGP is not SM but can result in various routing instabilities (e.g., permanent route oscillations). Enforcing compliance of BGP with Condition 5 is more difficult to achieve since even the first attribute of the BGP best path selection algorithm, i.e., the BGP local preference, is not M nor SM. As such, BGP can not be rendered compliant with Condition 5 by simply discarding specific options. Because the derived results (Section 5) do not directly apply to BGP, we handle this routing protocol as a special case by imposing the following restrictions:

BGP-specific Restrictions:

1. *BGP routes are selected only if no route is offered by other protocols.*
2. *Routes from BGP cannot be redistributed into a non-BGP protocol instance.*

These restrictions enforce SM between BGP and non-BGP instances, and ultimately guarantee correct routing so long as
(1) the BGP configurations do not result in anomalies,
(2) non-BGP instances comply with Condition 5, and
(3) conversion functions satisfy Condition 4.

Although these restrictions do not exist today, they do not prevent common design objectives from being accomplished: The first restriction makes BGP routes the least prefered routes. Indeed, when a router receives routes both from BGP and an IGP to a same destination prefix (e.g., 128.2.1/24), the router should typically prefer the more direct internal route learned from the IGP, to the external BGP route. This is because sending traffic through external networks (e.g., providers) can cost money. The second restriction prevents redistribution from BGP into IGPs. The question is whether this restriction would prevent existing design objectives from being achieved. Empirical studies [28] have found that network operators often inject routes from BGP into IGP, and this usage typically corresponds to VPN deployments: As illustrated in Figure 8, a company (e.g., XYZ) may have several sites (e.g., Site 1, Site 2) each deploying its own routing protocol (e.g., RIP, OSPF). To allow connectivity between the sites, the company relies on a service provider backbone. Routes from one site (e.g., OSPF routes from Site 1) are first redistributed into the backbone (i.e., BGP cloud) at an provider edge (PE) router (e.g., PE 1). The routes are then propagated through the BGP backbone, and finally redistributed from BGP into the IGP of each remote site (e.g., RIP from Site 2) at the connecting PEs (e.g., PE 2).

The fact that the new primitives prevent this type of redistribution may therefore seem a serious impediment to its adoption. However, it turns out that the same objective can naturally be achieved without any redistribution from BGP into IGP. For simple scenarios, a customer edge (CE) router (e.g., CE1) can originate a default route in the respective site's IGP (e.g., Site 1's OSPF), and

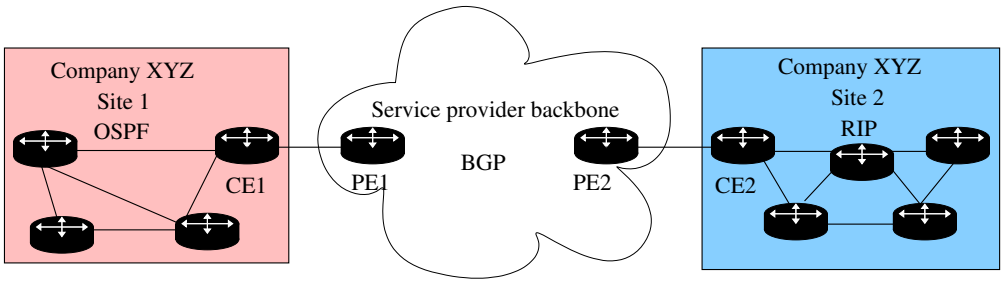

Figure 8: Illustration of typical VPN scenario.

be configured with a static route pointing to the connecting PE (e.g., PE 1) for the default route (0.0.0.0/0). As an alternative, BGP can also be deployed in the company's sites to propagate and filter the routes. In fact, BGP offers more flexibility and permits more efficient ways to route traffic.

Universal metric space: With the exception of BGP which is handled as a special case for the reasons discussed above, the results from the theory directly apply to the interactions between all the other routing protocols (OSPF, IS-IS, RIP, EIGRP, static routes). We note that recent proposals [23] have suggested modifications to the BGP protocol that would guarantee important properties (e.g., SM), while at the same time, still support existing policies (e.g., customer, provider, peer relationships). Those modifications would eliminate the need to treat BGP separately. However, we assume the current BGP protocol in the rest of this paper.

All non-BGP routing protocols (OSPF, IS-IS, RIP, EIGRP, static routes) can be unified under the following 2-ary metric space: $\{$*type, cost*$\}$. Derived from the theory, we make conversion functions an *explicit component* of the design. We treat each non-BGP protocol as a 2-ary algebra, where the first attribute is the route *type* and the second attribute is the route *cost*. We define the following universal metric space for the design of conversion functions:

1. Type: $\Sigma_{0,1} = \{\text{A}, \text{B}, \text{C}\}$. The universal domain for the type consists of three permitted elements, and is totally ordered with type A being prefered to type B which is in turn prefered to type C.

2. Cost: $\Sigma_{0,2} = \{1, 2, 3, \ldots, 2^{32} - 1\}$. The universal domain for the cost consists of the set of integers from 1 to $2^{32} - 1$ and is totally ordered by the arithmetic operator $\leq$.

Conversion functions: When comparing routes received from different routing protocol instances, our design maps the type and cost of each route into the universal metric space according to the default conversion functions shown in Table 3[4]. It then ranks the routes based on their ordering in the universal domains. Since RIP does not define a route type, all RIP routes are effectively of the same type "RIP". The same applies to static routes. The default cost conversion functions (e.g., $x \rightarrow x^8$ for RIP) are designed to scale the metric space (e.g., 4-bit for RIP) of each protocol to the 32-bit space of the universal cost domain. For example, an OSPF route of type "intra-area" and

[4]Details for IS-IS, and prohibited signatures, are omitted because of its similarities with OSPF and for conciseness respectively.

Metric	Protocol	To universal domain $\Phi_{protocol \to 0}$	From universal domain $\Phi_{0 \to protocol}$
Type	OSPF	intra-area → A inter-area → B external type 1 → C	* → external type 1
	RIP	RIP → C	* → RIP
	EIGRP	internal → B external → C	* → external
	Static	static → C	* → static
Cost	OSPF	$x \to x^2$	$x \to$ ceiling($\sqrt{x}$)
	RIP	$x \to x^8$	$x \to$ ceiling($\sqrt[8]{x}$)
	EIGRP	$x \to x$	$x \to$ ceiling(x)
	Static	$x \to x$	$x \to$ ceiling(x)

Table 3: Default conversion functions. The symbol "*" represents any permitted value.

cost "30" would be mapped into type "A" and cost "900" in the universal metric space. Similarly, an EIGRP route of type "internal" and metric "65345" would be mapped into type "B" and cost "65345". Since type A routes are prefered over type B routes, the OSPF route would be prefered.

The conversion functions in the other direction (i.e., from universal metric space to a protocol specific metric space) are for route redistribution. For example, let us assume that the OSPF route in the example above is being redistributed into RIP. It would be given a RIP hop-count of 3 because ceiling($\sqrt[8]{900}$) = 3.

The default conversion functions comply to Condition 4. Network operators may customize the conversion functions based on operational objectives subject to constraints: (1) the new conversion functions comply with Condition 4 as defined in Section 4, and (2) two routing processes at border routers must be configured with the same conversion functions if they belong to the same routing instance.

Route Selection: Like before, each routing process first determines a best route within its own RIB. For example, among all the received BGP routes, the BGP best path selection algorithm would choose a single most prefered BGP route. We note that currently, routers can run at most one instance of RIP and BGP but can run multiple processes of OSPF and EIGRP. Consequently, after each routing process has determined its best route, a router obtains at most one BGP route, but may receive multiple OSPF routes, each from a different OSPF process. To select one among them for the Forwarding Information Base (FIB), a router applies the following ranking rules in our design:

Step 1. Protocol: Prefer non-BGP (i.e., EIGRP, OSPF, RIP, static) routes to BGP route.

Step 2. Type: If multiple non-BGP routes are available, prefer type *A* routes, then type *B* routes, and type *C* last.

Step 3. Cost: Among non-BGP routes of the preferred route type, prefer the route with the lowest cost.

```
1   interface ethernet 0
2    ip address 192.1.1.1 255.255.255.0
3   !
4   interface ethernet 1
5    ip address 192.1.2.1 255.255.255.0
6   !
7   router rip
8    network 192.1.1.0
9    f(2,x) = 4 × x
10   f-1(2,x) = ceiling(0.25 × x)
11  !
12  router ospf 100
13   network 192.1.2.0 255.255.255.0 area 0.0.0.0
14   redistribute rip
15  !
16  ip route 192.1.3.0 255.255.255.0 192.168.1.10 98769
```

Figure 9: Illustration of new design in form of IOS commands.

If only one route is in consideration and it is from BGP, the process stops after rule 1 and selects the BGP route. Otherwise, it follows the ordering in the 2-ary universal metric space. Again, *step 1* of the proposed route selection procedure is created to handle the special case of BGP and to enforce the previously discussed restrictions. Similarly, the following route redistribution procedure treats BGP differently.

Route Redistribution: The theory allows the redistribution to be performed in either a vectoring or a link-state manner. In this design, for brevity, we restrict route redistribution to the vectoring mode.

We disallow any redistribution from BGP into a non-BGP protocol instance as part of the BGP-specific restrictions defined in the beginning of this section. We allow great flexibility for redistribution into BGP since this is not part of the BGP-specific restrictions. When a route is redistributed into BGP, its BGP attributes (e.g., local preference, AS-PATH, MED, community, etc.) can be set to any value as long as they do not cause routing anomalies within BGP. This flexibility allows the new primitives to preserve the current levels of autonomy, expressiveness and privacy between BGP networks: Any policy (e.g., customer, provider, peer relationships) currently implemented between BGP networks can still be accomplished. In addition, networks administered by different authorities and connected through BGP do not need to share more information than today: In particular, they do not need to exchange information on the conversion functions.

For redistribution between non-BGP instances (e.g., from OSPF into RIP), the metrics of the redistributed routes are decided by the conversion functions.

Configuration commands: The new primtives replace the *distance* and *default-metric* commands with two commands for defining the conversion functions.

1. The `f(<i>,x)` and `f-1(<i>,x)` commands are optional and allow network operators to override the default conversion functions for a routing process. The parameter `<i>` can be either be ”1” or ”2”, corresponding to the “type” or “cost” dimension, respectively. The new conversion

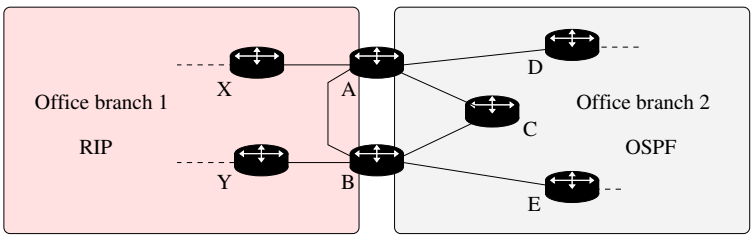

Figure 10: Illustration of domain backup.

functions must comply with Condition 4 as given in Section 4. We leave it to future work to develop an algorithm to automatically verify this requirement. Alternatively, a router vendor can restrict the choice of conversion functions to those known to comply. For example, for "cost", the conversion functions can be confined to the form $a \times x^b + c$ with specific ranges for the configurable parameters (a, b, c).

The design also requires modifications to the `redistribute` and `ip route` commands.

1. The `redistribute` command is simplified. All metric related options are removed.
2. The `ip route` command is enhanced to allow the assignment of a cost to a new static route.

Figure 9 illustrates a possible configuration with these commands. At lines 9-10, the conversion functions for RIP are customized. At line 16, a static route to destination prefix 192.1.3/24 is configured with a cost of 98769.

While we can formally establish that the presented design guarantees safety, the details are omitted for space reasons.

7 New primtives are expressive

In this paper, we define expressiveness broadly as the ability of the primitives to support operational goals. As expressed in Section 6, the current levels of autonomy, privacy and expressiveness between BGP networks are preserved. In the remainder of this section, we consider four operational objectives and examine how the new primitives may support them. The first three are considered important design goals by operators [28]. The last one illustrates the flexibility with which one can derive primitives from the proposed theory.

7.1 Domain backup

Domain backup designates the ability for a network to preserve reachability even in the event of a routing instance partition, through alternate physical paths traversing other routing instances. To illustrate the property, consider the network from Figure 10. It consists of two office branches, each running its own routing instance (RIP, OSPF). In the failure of router C, link A-C, or link B-C, the routers D and E can no longer directly communicate despite the existence of a physical

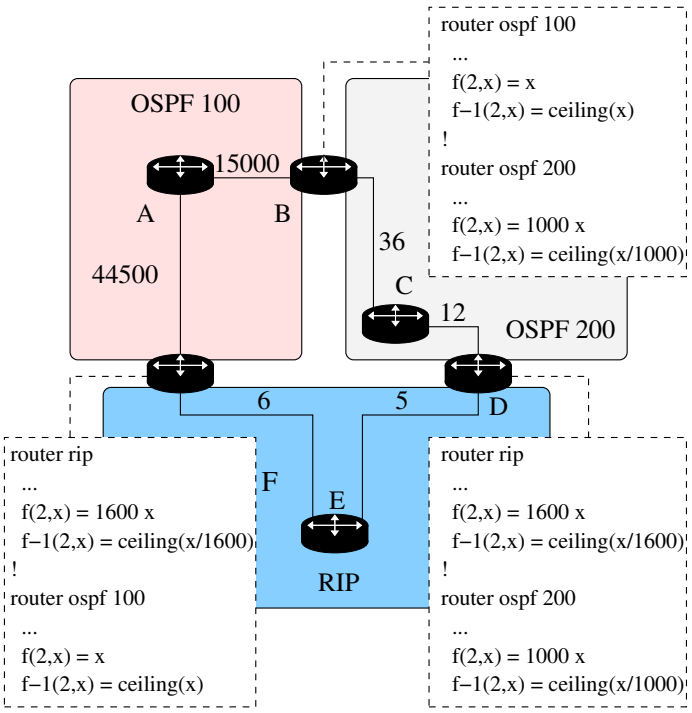

Figure 11: Illustration of router-level shortest path routing across OSPF and RIP instances.

path (D-A-B-E) between the two routers. By default, the path D-A-B-E is not offered as it traverses a different routing instance (RIP). To make this path available, mutual route redistribution should be enabled between OSPF and RIP at the border routers B and A, respectively. However, route redistribution at multiple points can easily result in routing anomalies [7]. Hence, to support domain backup, current route redistribution solutions require specific physical topologies and complex policies [28]: the routing instances must be connected in a star topology, and domain backup is provided *only* to the leaf routing instances.

In contrast, the new primitives can offer domain backup to every routing instance with no restriction on physical topology. For the scenario of Figure 10, a simple activation of mutual route redistribution between RIP and OSPF at both border routers A and B would suffice. In the absence of failure, E receives two paths to D: E-B-C-A-D and E-B-A-D. The former will be selected as it is an intra route (type A in the universal metric space) whereas the latter is external (type C). Then, In the failure of router C, link A-C, or link B-C, E still receives the path E-B-A-D. As such, routers D and E preserve their connectivity.

7.2 Router-level shortest path routing across IGP instances

Router-level shortest path routing across IGP instances designates the ability for a pair of end hosts in different IGP instances to route traffic to each other along the shortest path. Today, this property is supported but only between OSPF instances. IOS provides the option to preserve the cost of a route redistributed from one OSPF instance into another OSPF instance. However, the

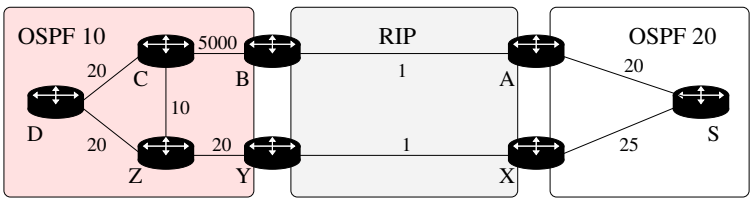

Figure 12: Illustration of traffic engineering across routing protocol instances.

current primtives do not permit router-level shortest path routing between two instances of different protocol types (e.g., OSPF and RIP). In such a setting, the cost of a redistributed route is set to a value with no relation to the cost of the initial route. The cost information to the destination is therefore lost at the first redistribution point. Even when redistributing between OSPF instances, the existing procedure has the following additional limitation. When a route from an OSPF instance (e.g., OSPF 100) is redistributed into a different OSPF instance (e.g., OSPF 200), the cost can only be set to either an arbitrary value that is independent from the initial cost (e.g., $Cost_{OSPF200} = 100$), or its original value (i.e., $Cost_{OSPF200} = Cost_{OSPF100}$). The current route redistribution procedure does not permit the cost of redistributed routes to be modified between the instances through a function (e.g., $Cost_{OSPF200}$ = 1600 $\times$ $Cost_{OSPF100}$). However, operational networks may rely on the OSPF cost to reflect the physical distance between routers. When a different unit is used in each instance (e.g., miles versus meters), router-level shortest path routing is not possible: the path with the lowest physical distance may not get selected.

In contrast, with the new primitives, operators can specify their own conversion functions overcoming the above limitations and enabling router-level shortest path routing between any pair of IGP instances. Figure 11 depicts an example of three routing protocol instances: OSPF 100, OSPF 200, and RIP. Their routing metrics model the physical distance in meters, kilometers and miles, respectively. The depicted conversion functions at the border routers (B, D, F) allow geographical shortest path routing across the three instances.

7.3 Traffic engineering

Current traffic engineering techniques are only applicable within one IGP routing protocol instance [14] or between BGP domains [34]. Our primitives, with their support for router level shortest path routing, naturally extends traffic engineering across multiple routing instances without requiring any additional coordination between the instances.

To illustrate the existing limitations and newly supported capabilities, we assume the network depicted in Figure 12. The network is composed of three routing protocol instances (OSPF 10, RIP, OSPF 20). Network operators frequently adjust the IGP weights to minimize congestion. The adjustments of IGP weights aim at redirecting traffic over less congested links. However, this technique is currently applicable only within a single routing protocol instance: We assume that the link B-C is congested and its weight is therefore updated to a larger value. The goal is for senders (e.g., S) to select the less resource constrained paths (e.g., S-X-Y-Z-D). However, because redistributed routes are assigned a static metric values (e.g., at B, Y, A and X), the initial

Metric	Protocol	To universal domain $\Phi_{protocol \to 0}$	From universal domain $\Phi_{0 \to protocol}$
Protocol	OSPF	OSPF $\to$ 254	* $\to$ OSPF
	RIP	RIP $\to$ 254	* $\to$ RIP
	EIGRP	EIGRP $\to$ 254	* $\to$ EIGRP
	static	static $\to$ 254	* $\to$ static

Table 4: Additional default conversion functions for the new *protocol* attribute.

weight information is lost and senders may still select the congested paths (e.g., S-A-B-C-D). Although the metrics of redistributed routes could be updated at the border routers B and Y in times of congestion, the network operators of OSPF 10 may have no control over the border routers A and X. As a consequence, congestion cannot be minimized across multiple routing instances. In comparison, our primtives' design does not have the same limitation. The default conversion functions (Section 6) would suffice in this case.

7.4 Strict preference policy

The current route selection allows routers to strictly prefer routes from one protocol instance over another, e.g., *"Always prefer OSPF routes to RIP routes"*. This type of policy might be useful to implement blackholes (e.g., in the event of DDoS). This section illustrates how our design can be extended to support such strict preference policy. Every non-BGP routing protocol instance is modeled as a 3-ary routing algebra: $\{$*protocol, type, cost*$\}$. The new *protocol* attribute is first considered when comparing non-BGP routes. Its has an integer range from 1 to 255 in the universal metric space, with 255 corresponding to the prohibited path. EIGRP, OSPF, RIP and static routes are defined to be of *protocol* type "EIGRP", "OSPF", "RIP" and "static", respectively. Table 4 presents the additional default conversion functions. All protocols are equally preferred by default. This design extension supports strict preference policies in addition to the previously presented objectives. To specificy a strict preference for a routing instance, a network operator simply overrides its conversion functions in the *protocol* dimension, e.g., from "OSPF $\to$ 254" to "OSPF $\to$ 10".

8 Implementation of the new primitives

We have implemented the new proposed primitives into the XORP routing software. Network operators can define their own conversion functions and the interactions between routing instances are carried through according to the new route selection and route redistribution procedures. This section gives a brief overview of the current implementation.

Default conversion functions are defined as suggested in Section 6. Network operators can override them to implement their own design objectives. When entering new values, the code verifies that the new definitions comply with the sufficient conditions for correctness. The parameters for the conversion functions can be specified either into a static configuration file – which is loaded at XORP startup time – or through the XORP command line interface.

For the *metric* attribute, the conversion functions from the routing protocol specific metric to universal units are restricted to the form of "$a \times x^n + b$" (with a, n, and b being the parameters), and the conversion functions from universal units are set to their inverse. This limitation permits a fast and easy verification of the conversion functions' compliance with the desired conditions, while still enabling all the operational goals presented in Section 7. The range of supported functions could be extended to support additional forms of functions. The conversion functions are currently defined through policy statements: e.g.,

```
policy{
     f_rip_2_universal_type:  c
     f_rip_2_universal_metric_a:  1
     f_rip_2_universal_metric_b:  0
     f_rip_2_universal_metric_n:  7
}
```

We note that the XORP configuration syntax resembles that of Juniper routers. As such, while the previous sections presented the new primitives in the form IOS configuration commands, the implementation shows that while the syntax may differ across platforms, the primitives' functionality still apply and can be implemented in different router platforms.

Although addressed by the theory and previous sections, the implementation does not support safe route selection and route redistribution with EIGRP, IS-IS and IPv6. EIGRP is Cisco proprietary and not supported by XORP. IS-IS is to be supported by XORP but not yet implemented as of XORP version 1.6. Finally, our implementation does not yet support route redistribution and route selection for IPv6 (e.g., OSPF version 3).

9 Future Work

Several important questions still need to be investigated. On the theory front, can we relax the requirement that routing processes at border routers must be configured with identical conversion functions when they belong to the same instance? In addition, can we tighten the sufficient conditions, especially to preserve associativity and isotonicity across multiple routing protocol instances? These two properties are particularly important when route redistribution is performed in a link-state mode to combine multiple link-state routing protocol instances into a super link state routing domain. The concept of merging distinct link state routing instances, while not feasible today, has potential operational benefits such as a more efficient use of resources through domain-wide traffic engineering and domain-wide back-up path planning.

On the design front, as raised in Section 7, what are all the important operational requirements for the primitives? How do we collect them and furthermore anticipate requirements that may arise in the future? Finally, on the operation front and specific to our design, what is the best strategy to verify that the set of conversion functions configured for a network conforms to the relevant safety conditions? We may restrict the choice of conversion functions to ones that are known to be safe. The main challenge then is to find a set of compliant functions with sufficient flexibility to support all the operation needs. Ultimately, we should develop algorithms that can derive a set of suitable conversion functions from given high level operational objectives.

10 Related work

A large body of work exists on the correctness of routing. Starting from the early ARPANET, researchers have looked into conditions that may impede the proper delivery of IP datagrams to their intended destinations [25]. However, most of prior work considered a specific protocol at a time. For RIP, the focus was on solving the "count to infinity" problem. For OSPF, special attention was given to its stability issues [4], [35]. For BGP, various causes for potential routing anomalies have been identified, followed by the development of thorough analytical models and solutions [26, 39, 33, 20, 38, 15, 21, 11, 31]. The insights gained from these efforts led researchers to explore design principles towards the creation of a safer inter-domain protocol [22, 24, 12, 13]. In addition, while algebraic structures have been proposed to solve a variety of network routing problems [6, 16, 17], recently, researchers have also relied on algebraic frameworks to identify fundamental properties a vector or link-state routing protocol must satisfy to ensure correct behaviors [36, 37, 23].

For routing across multiple routing protocol instances, several analytical models were recently introduced [29, 30], enabling rigorous analyses of the current design and exposing its deficiencies. These models also made the formulation of practical configuration guidelines possible [27]. However, this approach is inherently backward-looking: The models only apply to existing solutions, and the derived guidelines further restrict the expressiveness of the already rigid current primitives. In contrast, the theory we present in this paper is more general allowing a wide range of new designs. Indeed, we show that there exists safer and more expressive solutions than those used today. For example, as discussed in Section 3, while the current route redistribution procedure behaves like a vector protocol, our theory offers insights for safeguarding route redistribution in either a vectoring or a link-state manner.

11 Conclusion

We have presented, to the best of our knowledge, the first theory for reasoning about the safety of routing across multiple routing instances. The theory is general because it models the interconnections betwen any combination of link-state, distance-vector and path-vector routing protocol instances. In addition, we identify as set of conditions both for the routing protocol instances and the primitives to guarantee correct routing and optimal paths. The conditions not only permit the design of new safer and more expressive primitives, but can also guide in the design of new routing protocols.

The second part of the paper describes an application of the theory to create a new set of primitives that are much safer and more flexible than the currently deployed version. We assumed no changes to the specifications of the existing routing protocols, and we demonstrate that with very minimum changes to how they should be configured, new primitives can not only support existing operational objectives but also enable new functions that are important but not feasible today, all the while guaranteeing routing safety.

In the big picture, our effort can be viewed as another example that underscores the importance and feasibility of principled design in networking research. We believe there are basic sciences

behind all phases of network operations and hence, part of the current momentum in "green field" research ought to be directed toward developing fundamental theories.

12 Acknowledgement

Yi Zhuang and Aditya Bhave implemented the new primitives into the XORP routing software.

References

[1] 100x100 Clean Slate Project. www.100x100network.org.

[2] 4D Project. www.cs.cmu.edu/∼4D.

[3] XORP: eXtensible Open source Routing Platform. www.xorp.org.

[4] Anindya Basu and Jon G. Riecke. Stability Issues in OSPF Routing. In *Proc. ACM SIGCOMM*, 2001.

[5] Theophilus Benson, Aditya Akella, and David Maltz. Unraveling the complexity of network management. In *USENIX Symposium on Networked Systems Design and Implementation*, 2009.

[6] Bernard Carré. *Graphs and Networks*. Oxford University Press, 1979.

[7] Cisco. OSPF Redistribution Among Different OSPF Processes, 2006.

[8] Cisco. Redistributing Routing Protocols, 2006.

[9] Cisco. What is administrative distance?, March 2006.

[10] Cisco. Route Selection in Cisco Routers, 2008.

[11] Cheng Tien Ee, Vijay Ramachandran, Byung-Gon Chun, Kaushik Lakshminarayanan, and Scott Shenker. Resolving inter-domain policy disputes. In *Proc. ACM SIGCOMM*, 2007.

[12] N. Feamster, H. Balakrishnan, and J. Rexford. Some foundational problems in interdomain routing. In *Proc. ACM SIGCOMM Workshop on Hot Topics in Networking (HotNets-III)*, 2004.

[13] Nick Feamster, Ramesh Johari, and Hari Balakrishnan. Implications of autonomy for the expressiveness of policy routing. In *Proc. ACM SIGCOMM*, 2005.

[14] Bernard Fortz, Jennifer Rexford, and Mikkel Thorup. Traffic engineering with traditional IP routing protocols. In *IEEE Communication Magazine*, 2002.

[15] L. Gao and J. Rexford. Stable internet routing without global coordination. In *Proc. ACM SIGMETRICS*, 2000.

[16] M. Gondran and M. Minoux. *Graphs and Algorithms*. Wiley, 1984.

[17] M. Gondran and M. Minoux. *Graphs, Dioids, and Semirings : New Models and Algorithms*. Springer, 2008.

[18] M.G. Gouda and M. Schneider. Maximizable routing metrics. In *IEEE/ACM Transactions on Networking*, 2003.

[19] Mohamed G. Gouda and Marco Schneider. Maximizable routing metrics. In *Proc. IEEE ICNP*, 1998.

[20] T. Griffin and G. Wilfong. On the Correctness of IBGP Configuration. In *Proc. ACM SIGCOMM*, 2002.

[21] Timothy Griffin, F. Bruce Shepherd, and Gordon T. Wilfong. The stable paths problem and interdomain routing. *IEEE/ACM Trans. Netw.*, 2002.

[22] Timothy G. Griffin, Aaron D. Jaggard, and Vijay Ramachandran. Design Principles of Policy Languages for Path Vector Protocols. In *Proc. ACM SIGCOMM*, 2003.

[23] Timothy G. Griffin and Joao Luis Sobrinho. Metarouting. In *Proc. ACM SIGCOMM*, 2005.

[24] Aaron D. Jaggard and Vijay Ramachandran. Robustness of Class-Based Path-Vector Systems. In *Proc. IEEE ICNP*, 2004.

[25] A. Khanna and J. Zinky. The Revised ARPANET Routing Metric. In *Proc. ACM SIGCOMM*, 1989.

[26] C. Labovitz, G. R. Malan, and F. Jahanian. Internet Routing Instability. In *Proc. ACM SIGCOMM*, 1997.

[27] Franck Le and Geoffrey Xie. On Guidelines for Safe Route Redistributions. In *Proc. ACM INM Workshop*, 2007.

[28] Franck Le, Geoffrey Xie, Dan Pei, Jia Wang, and Hui Zhang. Shedding Light on the Glue Logic of the Internet Routing Architecture. In *Proc. ACM SIGCOMM*, 2008.

[29] Franck Le, Geoffrey Xie, and Hui Zhang. Understanding Route Redistribution. In *Proc. IEEE ICNP*, 2007.

[30] Franck Le, Geoffrey Xie, and Hui Zhang. Instability Free Routing: Beyond One Protocol Instance. In *Proc. ACM CoNEXT*, 2008.

[31] Y. Liao, L. Gao, R. Guerin, and Z.-L. Zhang. Reliable Interdomain Routing Through Multiple Complementary Routing Processes. In *Proc. ACM ReArch*, 2008.

[32] David Maltz, Geoff Xie, Jibin Zhan, Hui Zhang, Gisli Hjalmtysson, and Albert Greenberg. Routing design in operational networks: A look from the inside. In *Proc. ACM SIGCOMM*, 2004.

[33] D. McPherson, V. Gill, D. Walton, and A. Retana. Border Gateway Protocol (BGP) Persistent Route Oscillation Condition, 2002. Request for Comments 3345.

[34] B. Quoitin, C. Pelsser, L. Swinnen, O. Bonaventure, and S Uhlig. Interdomain traffic engineering with BGP. In *IEEE Communication Magazine*, 2003.

[35] A. Shaikh, C. Isett, A. Greenberg, M. Roughan, and J. Gottlieb. A Case Study of OSPF Behavior in a Large Enterprise Network. In *Proc. IMW*, 2002.

[36] Joao Sobrinho. Algebra and Algorithms for QoS Path Computation and Hop-by-Hop Routing in the Internet. In *Proc. IEEE Infocom*, 2001.

[37] Joao Luis Sobrinho. Network routing with path vector protocols: Theory and applications. In *Proc. ACM SIGCOMM*, 2003.

[38] R. Teixeira, A. Shaikh, T. Griffin, and J. Rexford. Dynamics of hot-potato routing in IP networks. In *Proc. ACM SIGMETRICS*, 2004.

[39] K. Varadhan, R. Govindan, and D. Estrin. Persistent Route Oscillations in Inter-domain Routing. In *Computer Networks*, 2000.