2008-12

Instability Free Routing: Beyond One Protocol Instance

Le, F.; Zhang, H.; Xie, Geoffrey

Instability Free Routing: Beyond One Protocol Instance

Franck Le
Carnegie Mellon University
franckle@cmu.edu

Geoffrey G. Xie
Naval Postgraduate School
xie@nps.edu

Hui Zhang
Carnegie Mellon University
hzhang@cs.cmu.edu

ABSTRACT

Today, a large body of research exists regarding the correctness of routing protocols. However, many reported global disruptions of Internet connectivity, e.g., inter-AS persistent loops, cannot be explained by looking at a single routing protocol at a time. In fact, these anomalies have long been suspected in the operator community to be caused by the interactions between routing protocols. The interactions between protocol instances are governed by two procedures at the border routers: route selection (RS) ranks routes from different protocol instances; and route redistribution (RR) exchanges routes between protocol instances. Prior studies hypothesized that RR may be responsible for a portion of the observed anomalies. In this paper, we provide analytical and experimental results to link RS, RR, and their interplay to anomalies discovered in operational networks. We show that RS by itself can cause route oscillations and loops, and that in all Cisco, Quagga, and XORP implementations, nondeterministic behaviors may occur because of their incorrect modeling of the dependencies between RS and RR. We identify the root cause for each of the instabilities and derive a configuration guideline as well as a functional model to eliminate them.

1. INTRODUCTION

One of the primary goals of a network is to ensure the proper delivery of packets to the intended destinations. Routing protocols play an essential role toward that objective. They disseminate routing information and allow routers to compute their forwarding tables. Because of their importance, a large body of research has been devoted to the correctness of routing protocols.

However, existing analytical frameworks as well as empirical studies for understanding routing dynamics concentrate on one routing protocol at a time, most notably BGP [22], [33], [19], [6], [18], [16]. The reality is that a large number of the reported disruptions of Internet connectivity, such as the ones listed below, cannot be easily explained by the misbehavior of a single routing protocol.

Persistent forwarding loops: Several studies [31], [34] have reported the existence of persistent forwarding loops *within an AS* or *across* multiple networks. The discoveries of inter-AS routing loops were surprising given the fact that BGP has been specifically designed to avert the formation of such loops via checking the AS PATH attribute. Those studies [31], [34] conjectured that the interactions between static routes and BGP may have originated the inter-AS loops. Our own discussions with operators reveal that the operational community also views the interactions between BGP and IGPs as a possible root cause of this problem: the injection of routes from BGP into IGP, and then re-injection of the same routes from IGP back into BGP at another location will reset the AS PATH attribute and render the BGP's guard against inter-AS loops ineffective.

IP prefix hijacks: Prefix hijacks, such as the notorious AS 7007 incident [30], periodically happen in the Internet. This type of anomaly can have a large impact, disrupting the connectivity of thousands of networks. Misconfigurations are frequently cited as the source of the problem. McPherson [29] took a closer look at the root cause of a recent IP prefix hijack and hypothesized that the interactions between BGP and static routes may have been at the origin of the routing anomaly. Our discussions with operators suggest that a sequence of route injections between BGP and IGPs, which resets the AS PATH, may also cause a network to advertise a prefix it does not own. Traffic for the affected prefix is subsequently black-holed when the offending network has insufficient resources to handle the increased amount of traffic.

Nondeterministic forwarding paths: Messages posted on bulletin boards used by network operators indicate that a network configuration consisting of multiple routing domains may result in unexpected forwarding paths. Chen and Yuan [7] described a case involving the interactions between iBGP and static routes. Our experiments show that the scope of the problem is significantly larger: e.g., interactions be-

ACM CoNEXT 2008, December 10-12, 2008, Madrid, SPAIN

tween BGP and OSPF, or between OSPF and RIP, have the same problem. None of the existing analytical frameworks can explain the observed outcomes.

The operational community has long suspected that the culprit of these anomalies may lie in the *interactions* between routing protocols. However, despite the severity of the problems, there is a surprisingly small number of studies on such interactions from the research community. Two of them [20], [32] introduced several frameworks to analyze the impact of the underlying IGP on BGP, and formulated conditions that may lead to forwarding loops, route oscillations, and delayed convergence. In our previous work [27], we developed a formal model to reason about the consequences of injecting routes across routing domains. In particular, that model can be used to explain the permanent inter-AS forwarding loops and IP prefix hijacks mentioned above.

Yet, these studies cannot make sense of all the anomalies described above. For example, none of them can provide a good explanation of the reported nondeterministic routing behaviors. In addition, the results in [20] and [32] are specific to the interactions between BGP and an IGP. However, recent empirical studies [28], [25] show that operational networks frequently deploy multiple instances of IGP and join them through route redistributions rather than with BGP to achieve important design objectives such as efficient routing. The interactions between these IGP instances require further research.

The interactions between different instances of routing protocols configured on one network, which we will simply refer to as *routing instances*, are currently governed by two procedures executed at border routers: route selection and route redistribution. The route selection procedure ranks routes received from different routing protocol instances and selects a "best" route among them for forwarding purposes, and the route redistribution procedure facilitates the exchange of routing information between routing instances. They are critically important for two reasons. First, they allow operators to fulfill a necessary function, that of integrating multiple routing domains. Second, operators make extensive use of route selection and route redistribution as primitives to achieve important design objectives that cannot be accomplished by routing protocols (including BGP) alone [25].

Prior studies [11], [12], [27] have provided some evidence based on simple scenarios that route redistribution is a separate risk factor from routing protocols for routing anomalies. In this paper, we present analytical and experimental results to link route selection and route redistribution to *reported* routing instabilities in operational networks, including the aforementioned forwarding loops and nondeterministic forwarding paths. We consider this a primary contribution of the paper. Additional contributions from this paper are as follows:

1. We show that *the problem is more fundamental* than previously reported. We show that route selection by itself – i.e., merely the presence of multiple routing instances in one network, without any exchange of routes between the routing instances – can also result in some of the reported routing anomalies.
2. We show that *the problem is broader* than previously reported. We present experimental results showing that all tested Cisco, Quagga, and XORP products have incorrectly implemented the dependencies between route selection and route redistribution which can cause nondeterministic routing outcomes.
3. We conduct *a root cause analysis* of the disclosed instabilities. We identify necessary conditions for each category of anomalies (loops, oscillations, nondeterministic routing behaviors) that can derive from route selection and its interplay with route redistribution. Our analysis indicates that the nondeterministic routing outcomes likely result from a lack of a detailed functional model of the dependencies between route selection and route redistribution.
4. Finally, we propose *a configuration guideline* to mitigate some of the instabilities. We formally prove that the guideline will prevent the targeted instabilities. We also present a functional model to precisely define the dependencies between route selection and route redistribution.

The rest of the paper is organized as follows. Section 2 provides more details on how the route selection and route redistribution procedures work and describes two key properties of their functionality. Section 3 analyzes the routing anomalies due to route selection. Section 4 addresses the additional instabilities caused by the interplay between route selection and route redistribution. Section 5 presents related work and finally, Section 6 concludes and discusses future work.

2. ABSTRACTING THE INTERACTIONS

A router can run multiple routing protocols (e.g., BGP, EIGRP, IS-IS, OSPF, RIP) at the same time. Certain vendors even allow a router to create multiple instances of the same routing protocol (e.g., OSPF 1, OSPF 2). A software process is associated with each of the created routing protocol instances and it is commonly referred to as a *routing process*. Each routing process is generally assigned a Routing Information Base (RIB) [13]. This database is used to store the routing information related to the routing process (e.g., routes received from peers).

2.1 Route Selection

A router may run multiple routing processes and receive more than one route (e.g., an OSPF route and a RIP route) to the same destination prefix. Some examples and motivations for such scenarios are described in Section 3. When receiving multiple routes to the same destination prefix from different routing processes, the router uses an inter-protocol route selection procedure to choose one of the routes to put

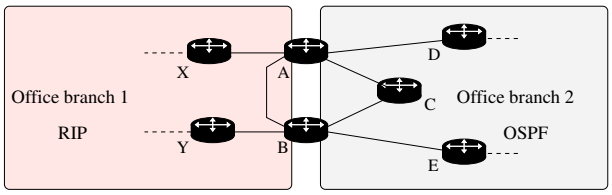

Figure 1: An enterprise with two office branches, each deploying its own routing protocol. By default, the RIP routers have no visibility of the destinations in the OSPF domain, and vice-versa.

in its Forwarding Information Base (FIB). This *route selection* procedure is the focus of our study. To add flexibility to the procedure, router vendors have introduced the concept of administrative distance (AD) [14] to aid ranking of routes from different routing protocols. Each routing process has a default AD value (e.g., 110 for OSPF and 120 for RIP on Cisco routers), which can be overridden per router and per prefix with special router configuration commands. All routes by default inherit the AD value of their respective routing process and the functionality of the route selection procedure can be precisely defined by the following property:

Route Selection Property (P1): *When multiple routing processes offer routes to the same destination prefix, the route with the lowest AD value is selected for the FIB.*

The routing process with the lowest AD value is referred to as the *selected routing process*, and the route that is put in the FIB (to forward traffic) the *active route*.

More specifically, each routing process first determines its best path using a protocol specific algorithm. For example, RIP prefers routes with the lowest metric value while BGP compares multiple criteria including the LOCAL_PREF, the AS PATH length and other parameters. Then, each routing process presents its most preferred route to the route selection procedure, which compares all the received routes and chooses the one with the lowest AD value.

To illustrate the route selection procedure, consider the network depicted in Figure 1. We focus on router A and we assume that it is configured with a static route to a destination prefix P. Router A runs a routing process of RIP and a routing process of OSPF, and we assume that both are configured with a lower AD value than that of the static route. When router A receives a route to destination P through a RIP neighbor, A shall prefer the RIP route to the static route and use it to forward the traffic.

2.2 Route Redistribution

Routing processes of different routing protocols by default are totally independent and do not exchange routing information even when they are running on the same router (e.g., OSPF process and RIP process on router A of Figure 1.) In fact, routing processes of the same routing protocol on the same router by default do not exchange routing information either (e.g., OSPF 1 and OSPF 2 on a same router). However, routing processes are required to exchange routing information with their peer processes, which are configured for the same routing protocol instance but on different routers (e.g., in Figure 1, RIP process on X and RIP process on A). More precisely, two routing processes are said to belong to the same *routing instance* when they run on different routers and form an adjacency, i.e., run the same routing protocol and exchange routing information.

When a network is composed of multiple routing instances, routes may need to be exchanged across routing instances. By default, routing information originated in a routing instance (i.e., by a member routing process) remains within the boundaries of that routing instance (i.e., shared only among routing processes of that routing instance). For example, in the network depicted in Figure 1, the RIP routers do not have visibility of the destinations in the OSPF instance and vice-versa. To allow communications across routing instances, vendors have introduced a router function called route redistribution, which must be explicitly enabled. The function can be enabled between any pair of routing processes (e.g., one RIP and the other OSPF) running on the same router to move routes from one (called source) into the other (called target). Although not formally specified by vendors, a key property for route redistribution is:

Route Redistribution Property (P2): *A route is advertised and redistributed only if it is active* [27].

A routing protocol should advertise a route only if the route is active. Violations of this property can result in routing anomalies as illustrated in Section 3.2. We note that by definition, link-state routing protocols relay all received routing information independently of whether a route is active. However, all vector protocols ought to satisfy property P2.

Following the same reasoning, a route should only be redistributed from a source routing process into a target routing process when the route is active. For example, consider a router running three routing processes u, v and w. Suppose that redistributions from u to v and from v to w are configured. In addition, assume that the active route has come from u. In such a case, the route is redistributed into v but not into w.

Finally, a recent study shows that operators use route selection and route redistribution to not only interconnect routing instances, but also meet important operational needs that cannot be provided by routing protocols alone [25].

3. INSTABILITIES OF ROUTE SELECTION

This section presents routing anomalies that can derive from route selection by itself, i.e., without any route redistribution configured between the routing instances. Section

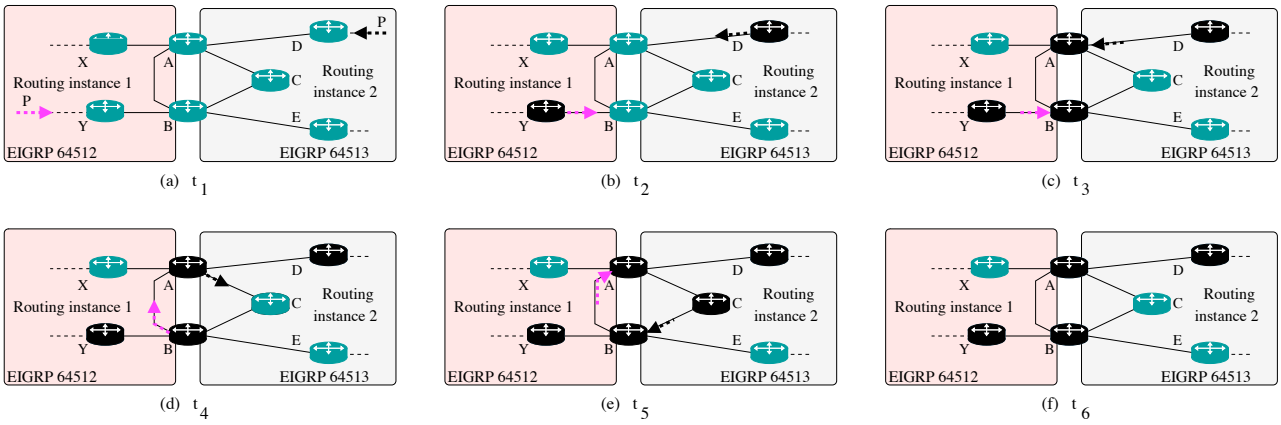

Figure 2: Illustration of route oscillations. The same destination prefix P is originated in both instances. The dashed arrows represent signaling messages. Routers shaded in dark represent routers with a route to the destination. The state at t_3 is identical to that at t_6.

3.1 illustrates how route oscillations and forwarding loops may occur. Section 3.2 analyzes the root cause for each of these anomalies. Finally, Section 3.3 presents a configuration guideline to eliminate these anomalies.

Topologies and routing designs similar to those described in this section have been observed in operational networks [28], [11], [25]. In addition, we have validated all the described scenarios, with the exception of those requiring specific race conditions, as it is difficult to create them. The validation environment consisted of Cisco 2600 routers (IOS Version 12.2).

3.1 Illustration of Routing Anomalies

We use the following notation throughout the paper. Routing instances are numbered 1, 2, ..., routers labeled A, B, ..., and routing processes denoted by <router>.<routing instance>. For example, B.1 designates the routing process from routing instance 1 at router B.

3.1.1 Route Oscillations

We assume the network depicted in Figure 2(a). It is similar to the network depicted in Figure 1 with the following difference: we assume that each office branch deploys an instance of a distance vector protocol, specifically EIGRP. Empirical studies have shown that operational networks commonly deploy multiple instances of a same routing protocol [11], [25]. We further assume that routers A and B are not configured to perform route redistribution, and have the default AD values for the two EIGRP routing processes. The discussion is in respect to a given destination prefix P originated by both instances. Recent empirical studies have also shown that in operational settings, one destination prefix can be originated by multiple instances [25]. P may be the default route, originated by each instance. Alternatively, each office branch may receive routes to the same destination through their respective ISP.

The following sequence of events illustrates the possible formation of a route oscillation.

t_1 Routes are propagated in both instances of EIGRP.

t_2 Upon receiving a route, routers D and Y learn a route to P and then further advertise the route to their neighbors (i.e., A and B respectively).

t_3 Router A receives a route from routing process A.2 pointing to D as the next-hop. Similarly, router B receives a route from routing process B.1 pointing to Y as the next-hop.

t_4 Router A further advertises the route into routing instance 2 through its neighbor router C. At the same time, router B further advertises the route into routing instance 1 through its neighbor router A.

t_5 Router A receives 2 routes (from A.1 and A.2). Because they have the same AD values, some implementations select the latest received information [15], i.e., the route from A.1. Similarly, router B receives 2 routes (from B.1 and B.2). Because they have the same AD values, B may select the route from B.2.

t_6 Since router A selected the route from A.1, A.2 stops advertising a route to P. In the same way, router B.1 stops advertising a route to P. Consequently, routers A and B lose their routes from A.1 and B.2 respectively. A reverts to using its previous route from A.2 and in the same manner, B uses the route from B.1 [13]. The resulting state is identical to that at t_4. In other words, we have a route oscillation.

The duration of the oscillation varies depending upon how long routing events at routers A and B are synchronized. Thus, such routing anomalies may have been diagnosed as transient forwarding loops. We note that the described scenario consists of only two routing instances. Studies [28],

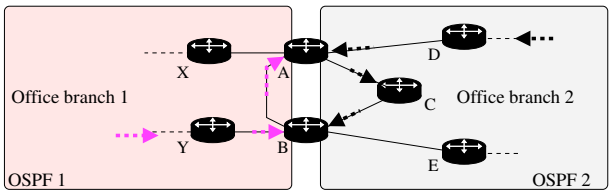

Figure 3: Illustration of a permanent forwarding loop (A-B-C-A). Routers A and B each receive two routes with identical AD values. The route selection is nondeterministic and if A selects the route from OSPF 1 and B, the route from OSPF 2, the loop results.

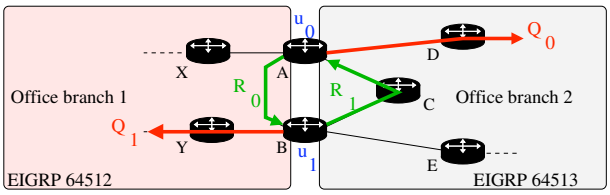

Figure 4: Illustration of the dispute wheel responsible for the oscillations observed in Figure 2.

[25] have disclosed that operational networks frequently deploy dozens or even hundreds of routing instances. In such large settings, the interactions would become significantly more complex and the chances for routing anomalies would increase considerably. In addition, the problem can be exacerbated by the proprietary nature of the AD concept: each router vendor has its own set of default values for routing protocols, and consequently, route selection configurations can result in not only oscillations of arbitrary time length but also *permanent* route oscillations [26].

While vendors have introduced proprietary solutions such as RIB quarantining [8] to mitigate the impact of route oscillations, it is necessary to fix the problems at their roots in order to restore connectivity. Prior studies of route oscillations have focused on BGP [6], [20], [19], [18]. We have just shown that the simple co-existence of IGP instances, even without BGP, can be another plausible cause.

3.1.2 *Forwarding Loops*

In addition to causing route oscillations, the route selections across multiple protocol instances can also result in permanent forwarding loops. This is the case when some of the protocol instances perform link-state routing. We assume the network depicted in Figure 3. The topology is identical to those previously considered. However, each of the office branches is now running OSPF. Empirical studies [25] provide evidence that operational networks deploy multiple instances of OSPF for administrative reasons: each office branch may be administered by a separate team. Such design may also result from a company merger or may be intentional in order to control the dissemination of the routes [11]. We further assume that no route redistribution is configured and all the routers use the default administrative distances for the routing processes.

As depicted in Figure 3, this configuration can result in a permanent forwarding loop (A-B-C-A): when router A receives two routes (from $A.1$ and $A.2$), because they have identical AD values, A may select the route from $A.1$ [11]. Similary, B may select the route from $B.2$ resulting in a forwarding loop A-B-C-A.

The depicted configuration nondeterministically results in a forwarding loop because of the random selection at the border routers, which increases the difficulties of the debugging task. This scenario shows that route selection by itself can be at the origin of observed permanent forwarding loops. In addition, $B.1$ (respectively $A.2$) may be configured with a strictly larger AD value than that of $B.2$ (respectively $A.1$). Such an AD assignment can happen in a multi-vendor environment, or result from a configuration error, and it will consistently result in a forwarding loop.

3.2 Analysis of Root Cause

Route oscillations: The route oscillations described previously (e.g., in Figure 2) occur because routers repeatedly advertise and withdraw a route. This happens in response to a prefered route being offered and then retracted. Since routers in a link-state protocol advertise all of their information – independently of their selected paths to a destination – the interactions of route selections between only link-state routing processes do not cause route oscillations. Route oscillations can occur only if the route selections involve a minimum of *two* routing protocol instances like BGP, RIP, and EIGRP, which only advertise routes that are active.

The root cause of the oscillations is in fact similar to those in the BGP context. Griffin et al. [18] demonstrated that the presence of *dispute wheels* can be responsible for route oscillations. Before presenting the formal definition of a dispute wheel, we first introduce some notations. [18] suggested the following definitions: Considering a simple, undirected graph, $G = (V, E)$, where V is the set of nodes and E the set of edges, we focus on a specific node, called the origin. Every other node attempts to establish a path to the origin. A path in G is either empty or a sequence of nodes $(u_k u_{k-1} ... u_0)$ such that for all $i \in [0, k-1]$, $\{u_{i+1}, u_i\} \in E$. Then, for every $u \in V$, $\mathcal{P}^u$ represents the set of permitted paths from u to the origin. Finally, for each $u \in V$, there is a ranking function λ^u, defined over $\mathcal{P}^u$: if $P_1, P_2 \in \mathcal{P}^u$ and $\lambda^u(P_1) < \lambda^u(P_2)$, then u prefers the path P_2 over P_1.

A dispute wheel $\Pi = (\vec{U}, \vec{Q}, \vec{R})$ of size k is defined as a sequence of nodes $\vec{U} = u_0, u_1, ..., u_{k-1}$, and sequences of non-empty paths $\vec{Q} = Q_0, Q_1, ..., Q_{k-1}$, $\vec{R} = R_0, R_1, ..., R_{k-1}$ such that for every $i \in [0, k-1]$, (1) R_i is a path from u_i to u_{i+1}; (2) $Q_i \in \mathcal{P}^{u_i}$; (3) $R_i Q_{i+1} \in \mathcal{P}^{u_i}$; (4) $\lambda^{u_i}(Q_i) \leq \lambda^{u_i}(R_i Q_{i+1})$. (All subscripts are modulo k.)

Figure 4 highlights the dispute wheel responsible for the route oscillations described in Section 3.1.1 and Figure 2.

Forwarding loops: The permanent forwarding loops occur because of deflections. Deflections are formally defined [20] as follows: Considering a destination router u_0, and assuming that u_n has selected the forwarding path $P(u_n) = u_n u_{n-1} \dots u_0$, $P(u_n)[u_i, u_j]$ denotes the subpath of $P(u_n)$ starting at u_i and ending at u_j with $i \geq j$. A deflection happens on $P(u_n)$ at u_i if $P(u_i) \neq P(u_n)[u_i, u_0]$ but for all $j > i$, $P(u_j) = P(u_n)[u_j, u_0]$.

It was shown that the occurrence of deflections between iBGP and its underlying IGP can result in forwarding loops [20]. We have just shown that deflections can result from the interaction between any two routing instances, causing permanent loops. In Figure 3, deflections at routers A and B are responsible for the observed loop. In fact, the forwarding loops occur because of the AD assignment at the different routers: the configuration includes a dispute wheel. Yet, as link-state protocols advertise all routes they receive, regardless if they are active, dispute wheels do not cause route oscillations but deflections.

3.3 Guideline for Safe Route Selection

Because the focus of this paper is on the interactions between routing protocols, we assume that packet forwarding within each routing instance is *free of instabilities*; more formally, the routing protocol converges and the forwarding paths for each destination form a directed acyclic graph where all routers of the routing instance are connected, and all the leaf node(s), i.e., node(s) with no outgoing edges, either are directly connected to the destination network or run multiple routing processes (i.e., serve as a border router joining multiple routing instances). Given a network, we consider all static routes across the routers to form a single routing instance and assume that this instance is also free of instabilities. Finally, we assume that routing instances are not deployed in an overlay fashion. A routing protocol instance k is deployed in overlay between two routers A, B when $A.k$ learns a route pointing to B as the next-hop but B is not directly connected to A, and A needs to rely on a routing instance instance different than k to reach B. While overlay networks (e.g., an iBGP mesh) can result in routing anomalies [20], [9], they are beyond the scope of this paper.

Griffin et al. [18] showed that the absence of dispute wheels guarantees the convergence of the route exchanges. This condition is a major result and has steered much of the recent research in BGP stability. However, although important, this result may not be practical especially for operators who need to configure a network with multiple IGP instances. The relationships between BGP networks (customer, provider, peer) form a hierarchy between them and guarantee the absence of dispute wheels [17]. However, routing protocol instances within a network do not present similar relationships nor patterns [25]. There is currently no guideline on how to assign the AD values to guarantee the safety of route selections. As such, we propose the following guideline.

Guideline 1: *For a destination prefix P, all processes of a routing instance shall share the same AD value and every routing instance shall be assigned a globally unique AD value.*

Theorem 1: *Guideline 1 guarantees the absence of dispute wheels spanning multiple routing instances and thus the convergence of the route selections.*

Proof: We prove it by contradiction. Assume that a network compliant with Guideline 1 still contains a dispute wheel $\Pi = (\vec{U}, \vec{Q}, \vec{R})$ of size k and spanning at least two distinct routing instances.

Step 1 By definition of the dispute wheel, each router u_i receives at least two paths (Q_i, and $R_i Q_{i+1}$) to the origin. Each of these paths may have been received through multiple routing processes. Considering all routing processes at u_i that offer the path $R_i Q_{i+1}$ (respectively, Q_i), let $u_i.\rho_i$ (respectively, $u_i.\rho_i'$) represent the routing process with the lowest AD value. By definition of the dispute wheel, Q_i is less preferred than $R_i Q_{i+1}$ at u_i. Let $AD(u_i.\rho_i)$ be the AD value of the routing process $u_i.\rho_i$. We derive that for all $i \in [0; k-1]$

$$AD(u_i.\rho_i') \quad \geq \quad AD(u_i.\rho_i) \tag{1}$$

Step 2 Because there is no configured route redistribution, and no routing instance is deployed in overlay, for two successive routers x and y on a forwarding path to the origin, the set of routing instances through which x learns the route is always a subset of those for y. As such, the fact that u_i learns $R_i Q_{i+1}$ from ρ_i implies that the router u_{i+1} (on the path $R_i Q_{i+1}$) is also running a routing process in ρ_i, and u_{i+1} learned the subpath Q_{i+1} from at least ρ_i. By definition of the dispute wheel, Q_{i+1} is less preferred than $R_{i+1} Q_{i+2}$ at u_{i+1}. Therefore, we derive that for all $i \in [0; k-1]$

$$AD(u_{i+1}.\rho_i) \quad \geq \quad AD(u_{i+1}.\rho_{i+1}) \tag{2}$$

Step 3 Since the network complies with Guideline 1, all routing processes within the same routing instance have the same AD value. Therefore, for every $i \in [0; k-1]$ (modulo k)

$$AD(u_i.\rho_i) \quad = \quad AD(u_{i+1}.\rho_i) \tag{3}$$

From equations (2) and (3), we derive

$$\begin{aligned} AD(u_1.\rho_0) &\geq AD(u_1.\rho_1) = \\ AD(u_2.\rho_1) &\geq AD(u_2.\rho_2) = \\ \dots &\geq \dots = \\ AD(u_0.\rho_{k-1}) &\geq AD(u_0.\rho_0) = AD(u_1.\rho_0) \end{aligned}$$

Since the network complies with Guideline 1, every routing instance is assigned a globally unique AD value. As such, from the previous equations, we derive

$$\rho_0 = \rho_1 = \dots = \rho_{k-1} \tag{4}$$

Step 4 From Steps 1 and 2, for all i, u_i learns the path Q_i from two routing instances: $\rho_{i'}$ and ρ_{i-1}. By definition of $\rho_{i'}$, we derive that

$$AD(u_i.\rho'_i) \leq AD(u_i.\rho_{i-1})$$

Then, from equation (4), since $\rho_{i-1} = \rho_i$, we obtain

$$AD(u_i.\rho'_i) \leq AD(u_i.\rho_i)$$

Finally, combining with equation (1), we conclude that

$$AD(u_i.\rho_i) \leq AD(u_i.\rho'_i) \leq AD(u_i.\rho_i)$$

which means that for all $i \in [0; k-1]$, $\rho_i = \rho'_i$. Therefore, $\rho_0 = \rho_1 = \ldots = \rho_{k-1} = \rho'_0 = \rho'_1 = \ldots = \rho'_{k-1}$. This contradicts the initial assumption that the dispute wheel spans multiple distinct routing instances. □

In addition to guaranteeing convergence, *Guideline 1* also guarantees loop-free forwarding paths.

Theorem 2: *Guideline 1 guarantees that the forwarding paths between the routing instances are devoid of permanent forwarding loops.*

Proof: Again by contradiction. Suppose a network compliant with *Guideline 1* contains a permanent loop $u_0u_1 \ldots u_{k-1}u_0$ for a destination prefix P. Let ρ_i denote the routing instance from which u_i learns its active route to P. Packet forwarding within each routing instance is free of instabilities. As such, there exists $i \in [0; k-1]$ such that u_i and u_{i+1} learn their active route to P from two distinct routing instances. Without loss of generality, we can assume that u_0 and u_1 learn their active route from two different routing instances ρ_0 and ρ_1. Note that the discussion below is with respect to destination P.

Step 1 This section assumes no configured route redistribution. Therefore, u_0 learns its active route from another member of routing instance ρ_0. This section also assumes that routing instances are not deployed in an overlay fashion. As such, the router to which u_0 forwards its traffic (i.e., u_1) is the router that advertised the route to u_0 through routing instance ρ_0. We conclude that u_1 is also a member of ρ_0. However, the active route at u_1 is learned from a different routing instance ρ_1. Consequently,

$$AD(u_1, \rho_0) \quad \geq \quad AD(u_1, \rho_1) \tag{5}$$

Step 2 For every $i \in [1; k-1]$, the router u_i points to u_{i+1} as its next-hop. As above, u_{i+1} is a member of ρ_i. Now, u_{i+1}'s active route can be learned from the same routing instance (i.e., $\rho_{i+1} = \rho_i$) or from a different routing instance ($\rho_{i+1} \neq \rho_i$). We will show that in both cases, we have

$$AD(u_{i+1}, \rho_i) \geq AD(u_{i+1}, \rho_{i+1}) \tag{6}$$

Case 1: $\rho_i = \rho_{i+1}$. Since the network complies to Guideline 1, all routing processes of a routing instance have the same AD value. We conclude that

$$AD(u_{i+1}, \rho_i) = AD(u_{i+1}, \rho_{i+1})$$

Case 2: $\rho_i \neq \rho_{i+1}$. Since u_{i+1} learns its active route from ρ_{i+1}, we derive

$$AD(u_{i+1}, \rho_i) \geq AD(u_{i+1}, \rho_{i+1})$$

Step 3 Because the network complies with Guideline 1, all routing processes within the same routing instance have the same AD value. In other words, for every $i \in [1; k]$ (modulo k)

$$AD(u_{i+1}, \rho_{i+1}) \quad = \quad AD(u_{i+2}, \rho_{i+1}) \tag{7}$$

From equations (5), (6) and (7),

$$\begin{aligned} AD(u_1, \rho_0) &\geq AD(u_1, \rho_1) = \\ AD(u_2, \rho_1) &\geq AD(u_2, \rho_2) = \\ \ldots &\geq \ldots = \\ AD(u_0, \rho_{k-1}) &\geq AD(u_0, \rho_0) = AD(u_1, \rho_0) \end{aligned}$$

In particular, $AD(u_1, \rho_0) = AD(u_1, \rho_1)$. As ρ_0 and ρ_1 are distinct, this equation contradicts Guideline 1 which states that every routing instance is assigned a globally unique AD value. □

Surprisingly, the proposed guideline has not been reported by the operational community despite its conceptual simplicity. It shows the value of the type of analysis carried out in this paper. It is unclear this guideline can accomodate all existing operational requirements. We leave this question to future work. Finally, in prior work [24], we derived similar guidelines for route redistribution.

4. INTERPLAY BETWEEN ROUTE SELECTION AND REDISTRIBUTION

The previous section disclosed routing anomalies caused by route selection alone, and identified a configuration guideline for safe route selection. This section analyzes new instabilities that can occur when route redistribution is used in conjunction with route selection. The focus is to explain why the interplay between route selection and route redistribution can easily cause the nondeterministic forwarding path problem described in Section 1.

Section 4.1 illustrates the anomaly, its severe consequences, and examines the extensiveness of the problem. We present experimental results which show that the problem is not just specific to one software implementation nor specific to one combination of routing protocols.

Section 4.2 hypothesizes possible causes for the observed anomaly based on additional experimental results. We postulate that the root of the problem is the lack of a precise specification of route selection and route redistribution, and more importantly, how the two procedures should interact.

Finally, Section 4.3 investigates how to eliminate the nondeterministic behaviors. We present a precise functional model

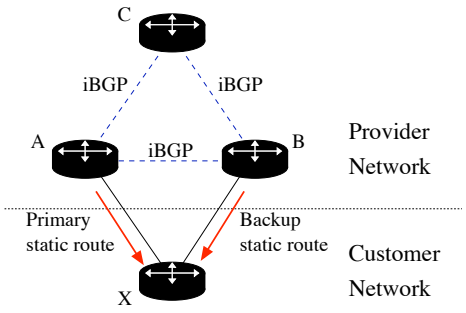

Figure 5: Nondeterministic forwarding paths.

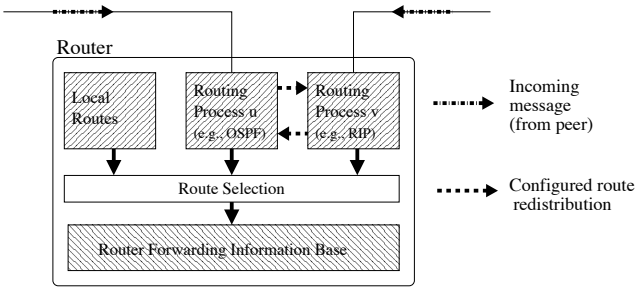

Figure 6: Experiment setup. A router is configured with two routing processes (u, v) and receives two routes to the same prefix. Mutual route redistribution is configured between u and v.

for guiding the implementation of route selection and route redistribution procedures. We formally establish the correctness and utility of the model by showing that an implementation compliant to this model will guarantee both of the essential properties defined in Section 2 at all times.

4.1 Nondeterministic Routing Behaviors

4.1.1 Motivating Scenario

The following scenario is first described in [7]. Consider the network depicted in Figure 5. It consists of a provider network offering Internet service to a customer network through two IP links: A-X and B-X. The routers A, B, and C in the provider network run an IGP and in addition, form a full iBGP mesh. At routers A and B, static routes pointing to prefixes inside the customer's network are redistributed into BGP so that they can be further propagated to other BGP networks. Suppose that the customer has designated the A-X link as the primary entryway for traffic arriving from the service provider, and B-X as a backup link. As such, the BGP process at router B is configured with a lower AD value (i.e., higher preference) than static routes. The expected behavior is that whenever the A-X link is up and A is reachable from B, B will forward all traffic to the customer network via A using the route offered from the BGP process.

However, it was reported that the forwarding paths at B surprisingly depend on the timing of when the static routes are entered at routers A and B [7]. Such a nondeterministic behavior is clearly an anomaly with severe consequences. Contrary to the design goal, B may forward traffic to the customer directly to X and announce this backup route to other BGP neighbors even though the primary link A-X is up and accessible.

To verify the report of [7] , we have implemented the topology of Figure 5 using 4 Cisco 3600 routers with IOS Version 12.2. We have observed the following behaviors at router B regarding a particular prefix in the customer network:

Case 1: When an iBGP route, redistributed from a static route that has been entered at router A, is the only route presented to the route selection procedure, it becomes the active route. Then, when a static route for the same prefix is installed locally at B, the iBGP route remains the active route because of its lower AD value. This is the expected and correct behavior.

Case 2: When the local static route is installed before the iBGP route from router A arrives, the static route becomes the active route and is locally redistributed into BGP. Then, when the iBGP route from A arrives, even though the newly received iBGP route has a lower AD value than the local static route, the static route remains the active route. This is an incorrect behavior because it violates property P1 as stipulated in Section 2.1. The route with a lower AD value did not become the active route.

4.1.2 Extensiveness of Problem

We have conducted more experiments to examine the extent of the nondeterministic behavior described above. We seek to determine whether the anomaly is specific to one software implementation (i.e., IOS 12.2) or one routing protocol (i.e., iBGP).

The experiments have a simple setup as depicted in Figure 6, where a single border router connects two routing instances. We have experimented with four different implementations for the border router: Cisco 3600 IOS version 12.2(24a), Quagga Software Routing Suite [4] version 0.98.6 (which is the latest stable release), Quagga Software Routing Suite version 0.99.10 (latest unstable release), and and XORP [5] version 1.4 (latest release at the time of the experiments).

For each implementation, two routing processes u and v are configured on the border router. One of the routing processes is configured with a lower AD value to provide the *primary* routes for all destinations. The other routing process should only provide *backup* routes. We have experimented with different protocol combinations for these processes. All the combinations are given in the first two columns of Table 1. Mutual route redistributions are configured between u and v.

In each experiment, we advertise two routes to a same destination prefix P to the border router. Static routes are directly entered to the router and the other routes are adver-

Source of Routes		Implementation			
Primary	**Backup**	**Cisco** (IOS 12.2)	**Quagga** (0.98.6)	**Quagga** (0.99.10)	**XORP** (1.4)
BGP	static	✗	✗	✗	✗
static	BGP	✓	✓	✓	✓
OSPF	static	✓	✓	✓	✓
static	OSPF	✓	✓	✓	✓
RIP	OSPF	✓	✗	✓	✗
OSPF	RIP	✓	✓	✓	✓
RIP	static	✓	✗	✓	✗
static	RIP	✓	✓	✓	✓
RIP	BGP	✓	✗	✓	✗
BGP	RIP	✗	✗	✗	✗
OSPF	BGP	✓	✓	✓	✓
BGP	OSPF	✗	✗	✗	✗

Table 1: Summary of experimental results. "✓" indicates a behavior conforming to property P1 (Section 2.1) and independent of the arrival order of the advertisements. "✗" signifies that the arrival order of the advertisements impacts the outcome of route selection.

tised from a neighboring router running a routing process that peers with either u or v. We vary the timing of the two route advertisements and then inspect the forwarding table of the border router to determine the outcome of the route selection and route redistribution procedures.

The results of the experiments are summarized in Table 1. The symbol "✓" indicates a behavior conforming to property P1 as stipulated in Section 2.1: the route with the lowest AD value is always selected as the active route, independent of the arrival order of the advertisements. The symbol "✗" signifies that the arrival order of the advertisements impacts the outcome of route selection and there are cases where the route with a higher AD becomes the active route.

We make the following two observations. First, all tested implementations produced unexpected outcomes some of the time. The problem therefore appears to be pervasive. Second, the outcome varied from implementation to implementation for some protocol combinations. This suggests that part of the problem may be due to software coding errors. For example, we discovered such an error in the Quagga version 0.98.6 source code: When a route is locally redistributed into the RIP protocol, all RIP messages received from the neighbors are in fact discarded independently of the AD values. This provides a good explanation of the observed outcomes with RIP when using the Quagga 0.98.6 implementation. However, given the pervasiveness of the problem, it seems more logical to conclude that the problem is not entirely due to implementation errors but comes from a lack of a precise model to understand, reason and support the interactions between route selection and route redistribution.

4.2 Analysis of Root Cause

It is difficult to pinpoint the root cause or causes of the observed anomalies because of the inaccessibility to the source code of the commercial implementations and the scarce documentation on this topic. In the following, we try to infer the root cause for the Cisco implementation indirectly from information available to us from the experiment described in Section 4.1.1.

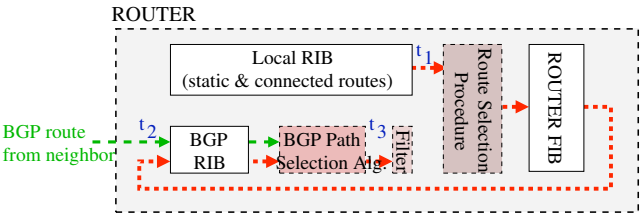

Figure 7: An incorrect model of the dependencies between RS and RR. A particular sequence of events would cause routes with higher AD values to be selected.

As suggested by [7], a look at the protocol specific routing information bases (RIBs) of router B shows that the redistributed local static route is present in the RIB of the BGP routing process. This discovery suggests that the Cisco implementation may have followed an incorrect model of the dependencies between route selection and route redistribution which is depicted in Figure 7. This model would cause a violation of route selection property (P1) for the scenario of Figure 5 under this sequence of events:

t_1 When the local static route is installed before the iBGP route from router A is received, the static route is selected to be the active route and redistributed into BGP.

t_2 When the iBGP route from A arrives at B, it is put into the same RIB as the redistributed local static route. This is confirmed by an inspection of the BGP RIB.

t_3 The BGP best path selection algorithm selects the best route among all the ones present in the BGP's RIB. By default, locally redistributed routes are assigned a WEIGHT value larger than that of routes received from BGP peers. WEIGHT is a Cisco-specific attribute [10] and for Cisco routers it is the first factor considered by the BGP best path selection algorithm. A route with a higher WEIGHT value is more preferred. Therefore in this case, the BGP best path selection algorithm selects the locally redistributed route as its best path. Then, for stability reasons, the BGP process will filter out the route since it was redistributed from another process to prevent it from being considered by the route selection procedure [27]. Finally, the process converges and the iBGP route from A, despite having a lower AD value than the local static route, does not become the active route at router B.

To confirm this conjecture, we reversed the WEIGHT values: locally redistributed routes are now assigned a lower WEIGHT value than the iBGP routes. We repeated the experiment, and this time the BGP selected the iBGP route as expected and the anomaly went away. However, this fix only applies to scenarios involving Cisco implementations of BGP. For example, it will not eliminate nondeterministic

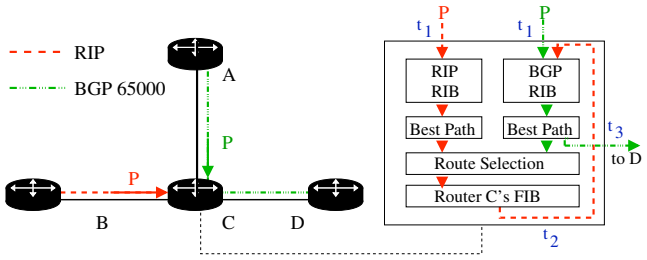

Figure 8: Existing implementations may cause a violation of Property P2 as stipulated in Section 2.2. Here, router C advertises a route that is not the active route for the prefix.

routing behaviors incurred by the interactions between RIP and OSPF.

The model depicted in Figure 7 can also fully explain the observed behavior when the iBGP route is received before the local static route is installed, i.e., Case 1 of Section 4.1.1. The iBGP route becomes the active route upon arrival. Later, when the static route is installed, both routes are considered by the route selection procedure. The iBGP route remains the active route because it presents a lower AD value. The static route is not redistributed to BGP because it is not the active route and the process converges as expected.

To summarize, we postulate that the nondeterministic routing behaviors are prevalent because of an incorrect functional model where locally redistributed routes are reconsidered as inputs to the routing protocol specific path selection algorithms (e.g., BGP best path selection algorithm). The negative impact of this kind of error is much bigger than a typical software bug. Next, we substantiate this point by showing that an implementation based on the incorrect model can cause additional unexpected outcomes by violating the Route Redistribution Property (P2).

Additional anomaly: Consider the network shown in Figure 8.

t_1 Router C receives two routes to the same prefix P from a BGP neighbor (A) and a RIP peer (B). Suppose that the route from RIP becomes the active route because the RIP process has been configured with a lower AD value.

t_2 We assume that redistribution from RIP into BGP is configured at C. As such, the active route from RIP is redistributed into BGP. The BGP RIB contains two routes to P: the locally redistributed route, and the BGP route from A. Suppose because of local policies (e.g., route-map setting a larger WEIGHT value to routes received from BGP neighbors) the BGP best path selection algorithm prefers the route from the BGP neighbor to the locally redistributed route.

t_3 As such, router C advertises the BGP route received from its BGP neighbor (A) to other BGP neighbors (e.g., D) instead of the active route, i.e., the locally redistributed route. This violates Property P2 and may cause deflections and permanent forwarding loops as illustrated in Section 3.2.

4.3 A New Functional Model Making Dependencies Unambiguous

This section presents a solution framework to eliminate the nondeterministic behaviors. The key element is a functional model of route selection and route redistribution that makes the dependencies between the two procedures unambiguous and guarantees both the route selection and route redistribution properties as defined in Section 2.

Section 4.3.1 describes a potential solution for vector protocols. Then, Section 4.3.2 extends the proposed solution to accommodate link-state protocols. The need for extension comes from the differences in these two types of routing protocols. While vector protocols first process the received information and only advertise the best paths, link-state routing protocols relay all the received information, even before computing the best paths. These characteristics require different designs. Finally, Section 4.3.3 shows that the proposed functional model guarantees the two properties given in Section 2.

4.3.1 A Functional Model for Vector Protocols

The proposed solution for vector protocols is depicted in Figure 9 (upper part). Each vector routing process (e.g., RIP, EIGRP) is assigned two RIBs: *RIBin* for incoming route announcements and *RIBout* for outgoing advertisements. A new announcement from a peer must first pass through some *filters*. The filters discard invalid advertisements and routes not compliant with local policies. For example, RIP routes whose metric exceeds 16 are filtered. After passing the filters, all routes are stored in the *RIBin*. A *protocol specific route determination algorithm* subsequently chooses the most preferred route among all routes to the same prefix.

Then, each routing process presents its best route to the *route selection procedure*. The active route is selected based on the AD values and installed in the *router's FIB*.

The *router's FIB* maintains the routes that are used to forward traffic. In this model, an active route is by default redistributed into the *RIBout* of the selected process. For example, if the active route comes from routing process $A.k$, then the active route is by default installed into the *RIBout* of routing process $A.k$ and advertised to the peer processes of $A.k$ in routing instance k. The active route may also be redistributed into other routing processes according to the route redistribution configuration on the router. Routing policies can be applied every time an active route is redistributed.

In this model, a locally redistributed route is not considered by any of the protocol specific route determination algorithms. As such, the status of this route is unambiguous from the perspective of the route selection procedure.

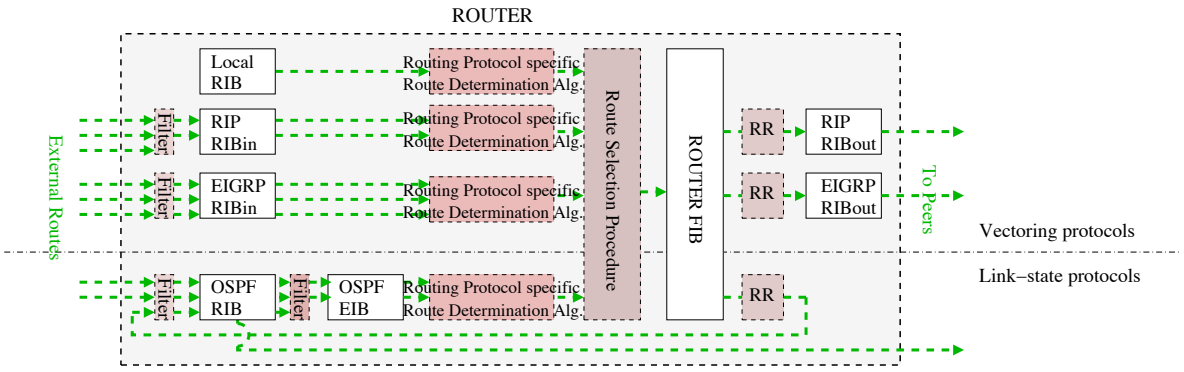

Figure 9: A functional model of RS and RR supporting all protocols.

4.3.2 Extension for Link State Protocols

This section extends the vector model to accommodate link-state protocols. As depicted in Figure 9 (lower part), each link-state routing process is also associated with two databases: a RIB and an *Eligible Information Base* (EIB). The RIB stores the regular link-state updates, including locally redistributed routes. All members of one link-state routing instance will eventually have identical information in their RIBs. Content-wise, EIB is a subset of RIB. It is a separate entity, used to isolate the routes that are eligible to become active at the router. An additional built-in filter between RIB and EIB prevents locally redistributed routes from entering the EIB. Then, the *protocol specific route determination algorithm* is executed based on the EIB and the best route is presented to the *route selection procedure*. Again, there is no ambiguity from the perspective of the route selection procedure.

The route redistribution part accordingly requires a simple extension. When the target routing process is a link-state protocol, the redistributed route is inserted into the RIB of the target routing process.

4.3.3 Correctness of the Proposed Model

The following theorem establishes the correctness of our model.

Theorem 3: *The proposed model guarantees route selection property P1 and route redistribution property P2 that are presented in Section 2.*

Proof: Consider a router and a destination prefix P. We first prove by induction that independent of the message arrival order, property P1 is guaranteed, i.e., the route with the lowest AD value is selected as the active route.
1. P1 is trivially satisfied when no route to P is offered.
2. We assume that at time t_1, the RS and RR procedures have converged and P1 holds true. Now, suppose the first routing event after t_1 occurs at time t_2: a route is withdrawn or a new route (either static or coming from a peer) is added. In the case of a route withdrawal, the RIBin(s) and the EIB(s) of the remaining routing processes that still have a route are not impacted by the route withdrawal. Each routing process selects its most preferred route and presents it to the route selection algorithm. The latter chooses the route with the lowest AD value. The active route may be redistributed into different routing processes, but this does not affect the RIBin(s) and the EIB(s). As such, the process converges and P1 remains true. In the case of a new route, contrary to existing implementations, locally redistributed routes are excluded from consideration by any of the protocol specific route determination algorithms. This eliminates the error condition described in Section 4.2. Consequently, each routing process that possesses a non empty set of routes to P presents its most preferred one to the route selection procedure, and the route with the lowest AD value is then selected to become the active route. Again, the active route may be redistributed into different routing processes, but this does not modify the content of the RIBin(s) and the EIB(s). As such, the process converges and P1 remains true.

We have shown that the model guarantees P1. Furthermore, in the proposed model, a vector routing process advertises routes, to its peers, from the RIBout. As such, a vector routing process advertises a route only if active. In addition, routes are redistributed directly from the router's FIB. Therefore, a route can be redistributed only if active. The model guarantees P2. □

5. RELATED WORK

Router vendors [13] mention that route selection can cause forwarding loops but do not provide any illustration nor guideline to avoid them. Some documents [12], [11], and [24] exposed instabilities due to route redistribution. In earlier work [27], we developed a framework to reason about the impacts of route redistribution at a network-wide level. Yet, this paper shows that route selection by itself, and its interplay with route redistribution, can also be the source of routing anomalies. Our work is the first to illustrate how routing instabilities may result from route selection alone and its interplay with route redistribution. We also analyze the root causes of these instabilities and develop guidelines and solutions for

preventing them.

Several studies [32], [20] looked at the interactions between BGP and its underlying IGP and revealed potential instabilities. In comparison to these studies, the scope of our work is much broader. Although our study does not encompass overlay routing protocols, we show that the interactions between any two routing processes, regardless which protocols they run, can create routing anomalies and the instabilities are not limited to route oscillations and loops.

6. CONCLUSION AND FUTURE WORK

We have demonstrated that the interactions between routing protocols are a much more complex problem than previously believed. While it has been recognized that route redistribution (RR) can easily cause routing anomalies and should be handled with care, this paper shows that route selection (RS) in itself, and its interplay with RR, can also result in a wide range of routing anomalies. It establishes a strong link between RS and RR and some of the puzzling routing anomalies discovered in operational networks.

The overall results suggest a twofold conclusion. On the one hand, the news is somewhat bleak. The RS and RR procedures are highly susceptible to routing anomalies and the range of anomalies is much wider than previously reported. Our study revealed that all tested implementations have incorrectly represented the dependencies between RS and RR. The lack of a well defined standard for these procedures has certainly compounded the problem. On the other hand, this paper shows that it might be possible to mitigate the instabilities through a deeper understanding of the problem. Many well-formulated theoretical frameworks have been developed for existing protocols, particularly for BGP. Because of its severity and prevalence, this problem deserves similar attention from the networking community.

In the big picture, we also see the need for ongoing efforts aimed at redesigning the Internet routing architecture [1], [23], [3] to closely examine the role of the interactions between routing instances. The correctness of individual routing protocols may still not be sufficient to guarantee correct routing in those settings. The current RS and RR procedures were invented without much consideration given to their safety properties. A clean slate redesign of these procedures, with an emphasis on robustness, should be highly desirable. [21] takes a first step in this direction. It introduces an elegant framework allowing operators to define and reason about routing protocols and their interactions. In the future, we hope to build upon such frameworks to design the next generation of "glue logic" [25] for routing protocols.

7. ACKNOWLEDGMENTS

We thank William Fischer, Rui Zhang-Shen and anonymous reviewers for their helpful suggestions. This research was sponsored by the NSF under the 100x100 Clean Slate Project [1] (NSF-0331653), the 4D Project [2] (NSF-0520187), grants CNS-0520210, CNS-0721574, and a Graduate Research Fellowship.

8. REFERENCES

[1] 100x100 Clean Slate Project. www.100x100network.org.
[2] 4D Project. www.cs.cmu.edu/∼4D.
[3] National Science Fundation NeTS Future Internet Design (FIND). www.nets-find.net.
[4] Quagga Software Routing Suite. www.quagga.net.
[5] XORP: eXtensible Open Router Platform. www.xorp.org.
[6] A. Basu, C.-H. L. Ong, A. Rasala, B. Shepherd, and G. Wilfong. Route oscillations in I-BGP with route reflection. In *Proc. ACM SIGCOMM*, 2002.
[7] E. Chen and J. Yuan. Deterministic Route Redistribution into BGP. Internet Draft, draft-chen-redist-00.txt, 2004.
[8] Cisco. Implementing RIB on Cisco IOS XR Software.
[9] Cisco. The "%TUN-5-RECURDOWN" Error Message and Flapping EIGRP/OSPF/BGP Neighbors Over a GRE Tunnel, 2005.
[10] Cisco. BGP Best Path Selection Algorithm, 2006.
[11] Cisco. OSPF Redistribution Among Different OSPF Processes, 2006.
[12] Cisco. Redistributing Routing Protocols, 2006.
[13] Cisco. Route Selection in Cisco Routers, 2006.
[14] Cisco. What is Administrative Distance?, 2006.
[15] Cisco. EIGRP Frequently Asked Questions, 2008.
[16] N. Feamster, H. Balakrishnan, and J. Rexford. Some Foundational Problems in Interdomain Routing. In *Proc. ACM SIGCOMM HotNets Workshop*, 2004.
[17] L. Gao and J. Rexford. Stable internet routing without global coordination. In *Proc. ACM SIGMETRICS*, 2000.
[18] T. Griffin, F. B. Shepherd, and G. T. Wilfong. The stable paths problem and interdomain routing. *IEEE/ACM Trans. Netw.*, 2002.
[19] T. Griffin and G. Wilfong. Analysis of the MED Oscillation Problem in BGP. In *Proc. ICNP*, 2002.
[20] T. Griffin and G. Wilfong. On the Correctness of iBGP Configuration. In *Proc. ACM SIGCOMM*, 2002.
[21] T. G. Griffin and J. L. Sobrinho. Metarouting. In *Proc. ACM SIGCOMM*, 2005.
[22] C. Labovitz, G. R. Malan, and F. Jahanian. Internet Routing Instability. In *Proc. SIGCOMM*, 1997.
[23] T. V. Lakshman, T. Nandagopal, R. Ramjee, K. Sabnani, and T. Woo. The SoftRouter architecture. In *Proc. ACM HotNets Workshop*, 2004.
[24] F. Le and G. Xie. On Guidelines for Safe Route Redistributions. In *Proc. ACM SIGCOMM INM Workshop*, 2007.
[25] F. Le, G. Xie, D. Pei, J. Wang, and H. Zhang. Shedding Light on the Glue Logic of the Internet Routing Architecture. In *Proc. ACM SIGCOMM*, 2008.
[26] F. Le, G. Xie, and H. Zhang. Instability Free routing: Beyond One Protocol Instance. Technical Report CMU-CS-08-123, May 2008.
[27] F. Le, G. G. Xie, and H. Zhang. Understanding Route Redistribution. In *Proc. IEEE ICNP*, 2007.
[28] D. Maltz, G. Xie, J. Zhan, H. Zhang, A. Greenberg, and G. Hjalmtysson. Routing design in operational networks: A look from the inside. In *Proc. ACM SIGCOMM*, 2004.
[29] D. McPherson. Internet Routing Insecurity::Pakistan Nukes YouTube? http://asert.arbornetworks.com/2008/02/internet-routing-insecuritypakistan-nukes-youtube.
[30] S. Misel. Wow, AS7007! www.merit.edu/mail.archives/nanog/1997-04/msg00340.html.
[31] V. Paxson. End-to-end routing behavior in the Internet. In *Proc. of ACM SIGCOMM*, 1996.
[32] R. Teixeira, A. Shaikh, T. Griffin, and J. Rexford. Dynamics of hot-potato routing in IP networks. In *Proc. ACM SIGMETRICS*, 2004.
[33] Varadhan, R. Govindan, and D. Estrin. Persistent Route Oscillations in Inter-domain Routing. In *Proc. Computer Networks*, 2000.
[34] J. Xia, L. Gao, and T. Fei. Flooding Attacks by Exploiting Persistent Forwarding Loops. In *Proc. of USENIX IMC*, 2005.