Instability Free Routing: Beyond One Protocol Instance

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

May 2008
CMU-CS-08-123

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.

Keywords: route selection, route reditribution, router configuration, routing anomalies

Abstract

The routing design of today's networks typically requires multiple instances of routing protocols to be configured. The interactions between the protocols are governed by two procedures at border routers: route selection ranks routes from different protocols; and route redistribution moves routes between protocols. The procedures are critical because operators rely on them to achieve important design objectives. However, there has been very little formal investigation into how safe they are. Existing analytical frameworks for studying routing dynamics have focused on individual routing protocols except for a recent paper that examines some anomalies caused by route redistribution. This paper presents the first comprehensive analysis of both route selection and route distribution regarding all three classes of routing instabilities: non-convergence, formation of loop, and non determinism. We show that the route selection procedure by itself can induce permanent route flaps and forwarding loops. We identify the necessary conditions or root causes for the instabilities and derive guidelines for eliminating them. We then present experimental results showing that all tested Cisco, Quagga, and XORP products have incorrectly implemented the dependency between route selection and route redistribution, causing non-deterministic outcomes. We address this problem by proposing a functional model that makes the dependency unambiguous.

1 Introduction

One of the primary goals of a network is to ensure the proper delivery of packets to the intended destinations. To achieve this objective, [10] identified the following three critical properties that the routing design of a network must satisfy:

- *Safety:* Given a set of routes and a set of policies, an assignment of the routes must exist such that no router wants to change its route in response to advertisements from other routers. Persistent route oscillations such as the ones caused by conflicting BGP policies [12] imply non-convergence and violate this property.

- *Validity:* The existence of a route to a destination implies that a packet sent along the corresponding path will eventually reach the intended destination. Forwarding loops and blackholes are examples of routing anomalies that violate this property.

- *Determinism:* Given a set of possible routes and a set of policies, the routers should always select the same predictable set of routes. This set of routes should be independent of the order in which the possible routes are propagated to the routers.

Because of their importance, a large body of research has been devoted to ensure that routing protocols satisfy these properties. However, most studies, particularly those proposing analytical frameworks, have focused on one single routing protocol at the time, despite that in reality multiple routing protocols are often used in the same network at the same time. There are growing evidences (e.g., the study of hot potato routing [19] and the work on iBGP [11]) that the interactions between concurrent routing protocols can also be a critical factor in determining a network's routing behaviors.

In the simplest scenario, a network deploys an IGP protocol (e.g., OSPF) for intra domain routing purposes and an EGP protocol (most likely BGP) to exchange routing information with other networks. Even in this basic setting, the IGP and the EGP protocols need to be interconnected. For example, some means are required to specify what routes from the IGP to advertise into the EGP and vice-versa. Recent empirical studies [17] revealed that the Internet routing landscape is in fact much more complex than the simple IGP/EGP setting. For example, dozens of distinct instances of routing protocols or routing domains may be present in one enterprise network and they have intricate interconnections throughout the network.

The interactions between routing protocols are governed by two software procedures running at border routers: the *route selection* procedure ranks routes received from different routing protocols and selects a "best" route among them for forwarding purposes, and the *route redistribution* procedure facilitates the exchange of routing information between protocols. These functions are critically important for two reasons. First, they allow operators to fulfill a necessary function, that of interconnecting routing protocols. 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 alone. Route selection and route redistribution are powerful tools that allow operators to implement a wide range of policies. (We give an example of such policies in Section 2.)

Despite their prevalence and critical role, the route selection and route redistribution procedures have in general received very little attention from the networking community. It was only recently that [15] discovered that route redistribution is vulnerable to routing anomalies similar to the policy oscillations in BGP.

This paper presents the first comprehensive analysis of both route selection and route distribution regarding all three classes of routing instabilities: non-convergence, formation of loop, and non-determinism. The major contributions are:

- We show that *the problem is more fundamental* than reported. We illustrate that the route selection procedure by itself – with no route redistribution enabled in the network – can induce permanent route flaps and forwarding loops. Route selection is a much more basic function than route redistribution because the former is invoked as long as multiple routing protocols are active at the same time while the latter must be enabled with additional configuration commands.

- We show that *the problem is much broader* than reported. We illustrate that the interplay between the route selection and route redistribution procedures can result in violations of all the forementioned safety, validity, and determinism properties. In particular, we present experimental results showing that all tested Cisco, Quagga, and XORP products have incorrectly implemented the dependencies between route selection and route redistribution, causing non-deterministic routing outcomes.

- We present *a comprehensive analysis* of all the instabilities. We identify and formulate the necessary conditions or root causes for each category of instabilities. We show that the complexity of determining if a given route selection configuration can result in forwarding loops is NP hard. Our analysis also 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.

- Finally, we propose a set of guidelines or solution framework to address all the instabilities. For each guideline, we formally prove that it will prevent the targeted instabilities. We also present a functional model that precisely model the dependencies between route selection and route redistribution and make them unambiguous.

The rest of the paper is organized as follows. Section 2 provides more details of how the route selection and route redistribution procedures work and describes two key properties of their functionality. Section 3 introduces some of the notation and more importantly, a couple of key assumptions for this work. Section 4 analyzes the routing anomalies due to route selection. Section 5 addresses the additional instabilities caused by the interplay between route selection and route redistribution. Section 6 presents related work and finally, Section 7 concludes and discusses future work.

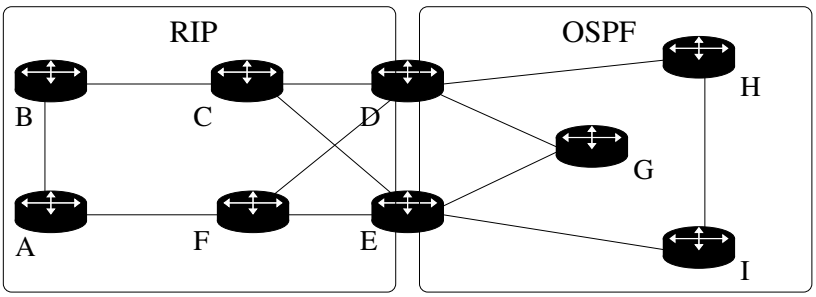

Figure 1: An enterprise with two office branches, each deploying its own routing protocol. By default, the RIP routers have no visibility of the destinations in the OSPF domain, and vice-versa.

2 Background

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) [7]. This database is used to store the routing information related to the routing process (e.g., routes received from peers).

Route Selection. A router that runs multiple routing processes may receive more than one route (e.g., an OSPF route and a RIP route) to the same destination prefix. When that happens, the router uses an inter-protocol route selection procedure to choose one of the routes to put 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) [8] 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 processes 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 PREFERENCE, 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 D and we assume that it is configured with a static route to a destination prefix P. Router D runs a RIP routing process and an OSPF process, and we assume that both are configured with a lower AD value than that of the static route. When router D receives a route to destination P through a RIP neighbor, D shall prefer the RIP route to the static route and use it to forward the traffic.

If multiple routing processes present the same lowest AD value, the router selects one of them according to a vendor specific algorithm. The selection may be random or the first received route may get selected. Within the selected routing process, we note that multiple routes may present an equal minimal cost (e.g., two OSPF routes may exist for the same destination with an identical cost). In such case, the router typically load balances the traffic on these equal cost routes.

Route Instance Abstraction [17]**.** 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 D 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., EIGRP 1 and EIGRP 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 C and RIP process on D). 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. Viewing networks at the routing instance level is useful in our analysis because it abstracts away many router level details that has little effect on network wide routing dynamics and more importantly allows us to focus on the interactions between different routing instances on a smaller set of routers.

Route Redistribution. When a network is composed of multiple routing instances, routes may also 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 at router configuration time. 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 [15]:

Route Redistribution Property (P2): *A route is redistributed only if it 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.

In fact, route selection and route redistribution are not only used to interconnect routing instances but also to achieve more complex functions that cannot be provided by routing protocols alone. In a network composed of multiple OSPF routing instances (e.g., because of a company merger), the operator may want shortest-path routing from any source to any destination across the network. BGP appears as a natural solution to interconnect the routing instances. Each office branch can be assigned a private BGP autonomous system (AS) number and BGP allows routing information to be exchanged across them. However, because the current BGP standard does contain any concept of link cost, it cannot support shortest-path routing across routing domains. [16] provides multiple scenarios illustrating the existing limitations in BGP to support efficient routing.

In comparison, route redistribution allows operators to achieve optimal routing across the routing instances by preserving the cost of a route when redistributing it from one instance of OSPF into another instance of OSPF [20]. Each router can then compute the global cost of the routes to a destination and select the shortest path.

Another common usage of route redistribution is to support partition healing. This property is also called domain backup and BGP does not offer it [18]. In the network from Figure 1, we assume that the links $<H,I>$ and $<D,G>$ fail. Routers H and I can no longer directly communicate. Although there are multiple physical paths between these two routers (e.g., H-D-F-E-I), if the two domains are each assigned a private BGP autonomous system (AS) number and interconnected through BGP, those paths will not be offered. This is because a BGP AS discards all advertisements with its own AS number in the AS PATH. Such behavior can be overridden in certain vendor equipments but BGP then becomes vulenarable to forwarding loops. Instead, route redistribution can safely support partition healing [15], [14].

3 Notation and Assumptions

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

Because the focus of this paper is on the interactions between routing protocols, we assume that packet forwarding with each routing instance is *free of instabilities*; more formally, i.e., the routing protocol converges and the forwarding paths for each destination network 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 the static routes across the routers to form a single routing instance and assume that this instance is also free of instabilities.

Finally, without loss of generality, all discussions are with respect to a single destination prefix, denoted by P, unless noted otherwise.

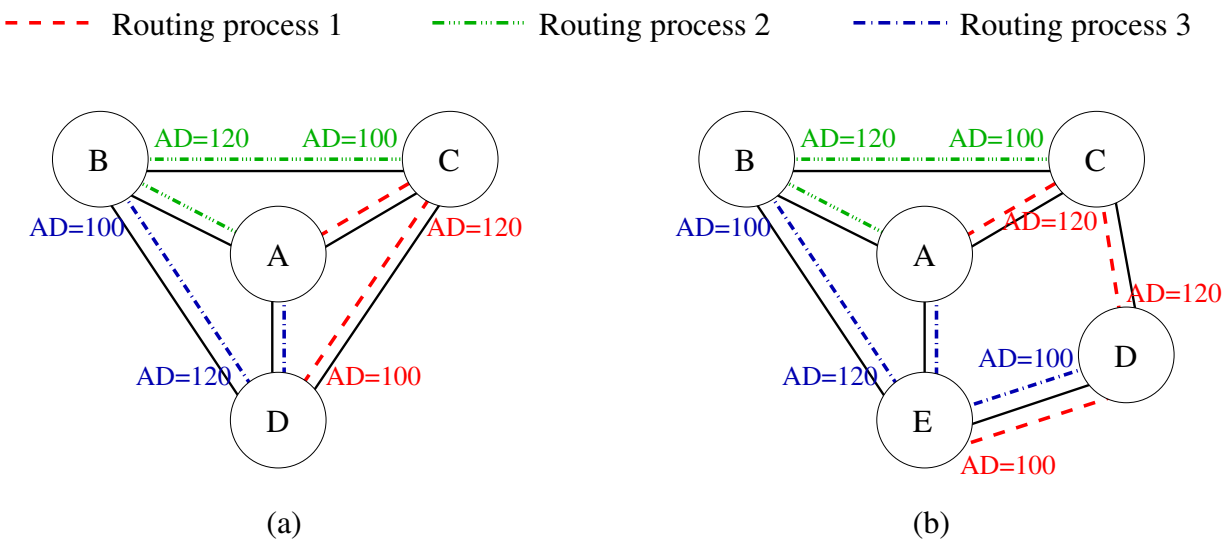

Figure 2: Illustration of route oscillations. The vertices (A, B, ...) represent the routers, and the solid edges between them indicate the physical connectivity. The small dots inside the vertices represent the routing processes, with their AD values annotated above them. Each dotted edge denotes an adjacency between a pair of routing processes.

4 Instabilities of Route Selection

For different factors, networks are frequently composed of multiple routing instances. In the simplest case, BGP Autonomous Systems (AS) deploy an IGP to exchange routing information within the AS and BGP to connect with other ASes. The interactions between routing processes can be complex and create forwarding loops as well as route oscillations. Section 4.1 addresses the occurrence of route oscillations and Section 4.2 focuses on the formation of forwarding loops.

This section assumes the usage of route selection solely. The next section addresses the additional routing anomalies that can arise when also considering route redistribution.

4.1 Route Oscillations

Section 4.1.1 illustrates the occurrence of route oscillations between multiple routing instances, Section 4.1.2 analyzes the root causes of these instabilities and Section 4.1.3 provides a sufficient condition for an oscillation-free configuration.

4.1.1 Illustration of Route Oscillations

We assume the network depicted in Figure 2(a). The scenario is inspired from [12]. The network may be an entreprise network with three office branches, each administered by a different team and running its own routing instance (1, 2, 3).

The network comprises 4 routers (A, B, C, D). Routers A, C, D belong to routing instance 1, routers A, B, C to 2 and routers A, B, D to 3. We focus on a destination prefix P originated by

router A and advertised in all three instances. The three routing instances are interconnected (e.g., to exchange certain routes not including P) and present the following preference order: Router B prefers routes from 3, C prefers routes from 2 and D prefers routes from 1. This order of preference may result from the default behaviors of routers in a multi-vendor environment. Each router vendor has defined its own preference order between routing protocols, and these orders differ across vendors.

The numbers located close to the routers represent the AD values at each border router. For example, routing process $D.1$ has an AD value of 100. As such, all routes received from routing process $D.1$ have an AD value of 100. The AD values of the routing processes at router A can be set to any arbitrary value and are therefore not represented in the figure.

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

t_0 At the initial state, we assume that none of the routers B, C nor D has a route to P.

t_1 Router A advertises a route to P in all three instances to neighbors B, C and D.

t_2 C receives a route to P from $C.1$. The route is the only option at C and therefore becomes the active route to P. Then, C further advertises the route to P in $C.1$. More specifically, C announces the route to D through routing instance 1. Similarly, B (resp., D) learns a route to P and further advertises it to C (resp., B) through routing instance 2 (resp., 3).

t_3 Routers B, C and D each receives two routes to P. D receives a route from $D.1$ and $D.3$. Because $D.1$ has a lower AD, D selects the route from $D.1$ and stops advertising P in $D.3$. Similarly, B (resp., C) selects the route from $B.3$ (resp., $C.2$) and stops announcing P in $B.2$ (resp., $C.1$).

t_4 Each router B, C and D now only receives a single route from A.

We note that the states at t_4 and t_1 are identical. Consequently, the routes oscillate between these states. We assumed a specific initial state and sequence of events. However, independently of the initial state and of the message arrival order, one can verify that the network of Figure 2(a) results in persistent route oscillations. We implemented the depicted topology with three instances of EIGRP (EIGRP 10, EIGRP 20 and EIGRP 30) and we observed the occurence of persistent oscillations.

Other configurations may only experience route oscillations depending on the initial state and the message arrival order. Figure 2(b) is an example. In some cases, the routes can converge. In other cases, the routes can oscillate for an arbitrary time length.

4.1.2 Analysis of Root Cause

A route oscillation occurs when a router repeatedly advertises and withdraws a route. This happens in response to a prefered route being offered and then retracted. Because routers in a link state protocol advertise all of its information – independently of its selected paths to a destination – the interaction between multiple link state routing processes do not cause route oscillations. Similarly, a network deploying a link state routing instance and a vector routing instance is safe from route

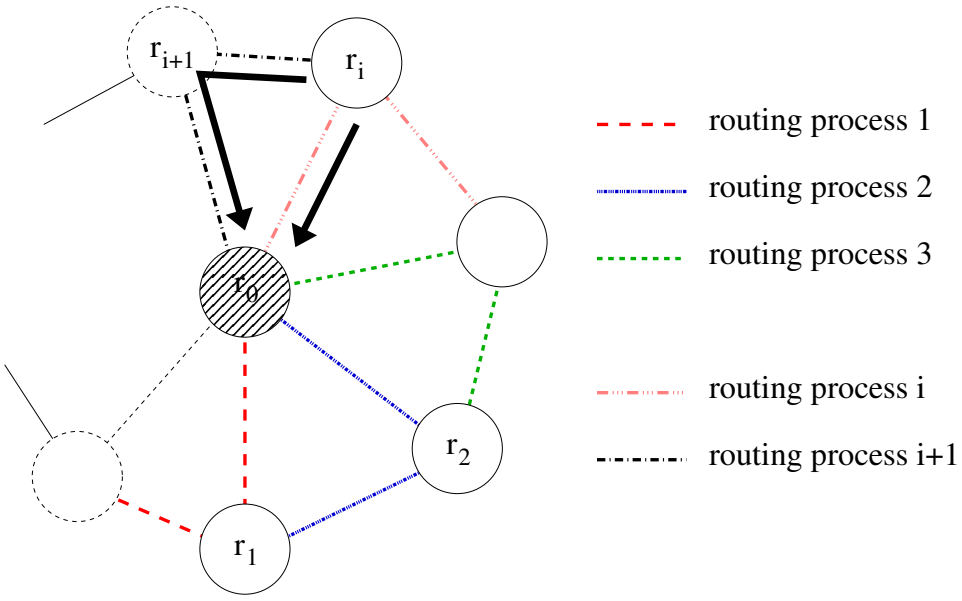

Figure 3: MP dispute wheel. (To avoid clutter, the routing processes and the AD values are not depicted).

oscillations between these two instances. Instead, route oscillations may appear when a router run multiple routing processes and each process only advertises a route when active. This is indeed the case with most vector protocols such as RIP and EIGRP.

The problem is in fact comparable to the emergence of route oscillations in the BGP context. [12] demonstrated that determining whether a stable path assignment exists is a NP-complete problem. Also, the presence of *dispute wheels* is a necessary condition for a network to diverge. Similarly, we define a *Multi Processes (MP) dispute wheel*.

First, let G = (V, E, R) be an undirected graph where V = $\{r_0, r_1, ..., r_m\}$ is the set of routers, R = $\{1, 2, ..., n\}$ the set of the routing instances and E the set of links between the routers.

A *MP dispute wheel* is a destination prefix P, a set of k ($k \leq n$) routers (r_1, r_2, ..., r_k) such that for all *i modulo k* ($1 \leq i \leq k$), (1) router r_i runs two routing processes $r_i.i$ and $r_i.(i+1)$ (2) routing process $r_i.i$ has a higher or equal AD value than that of routing process $r_i.(i+1)$ for destination prefix P, and (3) information received in routing process $r_i.(i+1)$ is originally announced by router r_{i+1}.

Figure 3 illustrates a MP dispute wheel. In the depicted configuration, r_0 participates in routing instances $1, 2, ..., k$, and originates a route to P in all of them. The physical links are omitted to reduce clutter. The thick solid arrows represent two forwarding paths router r_i may receive.

We observe that both configurations from Figure 2 contain a MP dispute wheel. In fact, we show that the presence of a MP dispute wheel is a necessary condition for route oscillations.

Theorem 3.1: *That the network contains a MP dispute wheel is a necessary condition for route oscillations between multiple routing instances.*

Proof The same reasoning than the one provided in [12] to demonstrate that the presence of

dispute wheels in BGP configurations is a necessary condition applies. □

4.1.3 Sufficient Condition for Convergence

From Theorem 3.1, we derive a sufficient condition guaranteeing that the interactions betwen the routing instances converge.

Theorem 3.2: *That the network is devoid of MP dispute wheels guarantees that the route selections at the different routers converge.*

Proof Because the presence of a MP dispute wheel is a necessary condition for route oscillations, the absence of MP dispute wheels guarantees the convergence of the routes exchanged between multiple routing instances. □

While the previous result is important, it may not be practical especially for an operator who needs to configure a network. The following guideline provides a mean to ensure that a network does not contain any MP dispute wheel.

Guideline 3.1: *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 3.3: *Guideline 3.1 guarantees that the route selections between the routing instances converge.*

Proof We consider a network G = (V, E, R) compliant with Guideline 3.1. We first show, by contradiction, that this network does not contain any MP dispute wheel. We assume the existence of a MP dispute wheel in G. Let P be a destination prefix and r_1, r_2, ..., r_k a set of routers in V such that for all *i modulo k* ($1 \leq i \leq k$), (1) router r_i runs two routing processes $r_i.i$ and $r_i.(i+1)$ (2) routing process $r_i.i$ has a higher or equal AD value than that of routing process $r_i.(i+1)$ for destination prefix P, and (3) information received in routing process $r_i.(i+1)$ is originally announced by router r_{i+1}.

For a destination prefix P, a router r and a routing instance i, we note $AD(i, r, P)$ the AD value of the route received by routing process i at router r for destination prefix P. The MP dispute wheel implies the following set of equations:

$AD(1, r_1, P) \geq AD(2, r_1, P)$,
$AD(2, r_2, P) \geq AD(3, r_2, P)$,
...,
$AD(k, r_k, P) \geq AD(1, r_k, P)$.

In addition, the network complying with Guideline 3.1, all routing processes within the same routing instance have the same AD value. In other words,

$AD(1, r_1, P) = AD(1, r_k, P)$,
$AD(2, r_2, P) = AD(2, r_1, P)$,
...,
$AD(k, r_k, P) = AD(k, r_1, P)$.

From these two sets of equations, we derive that:
$AD(1, r_1, P) \geq AD(2, r_1, P)$ =
$AD(2, r_2, P) \geq AD(3, r_2, P)$ = ... =
$AD(k, r_k, P) \geq AD(1, r_k, P) = AD(1, r_1, P)$.
which implies that
$AD(1, r_1, P) = AD(2, r_1, P)$ =
$AD(2, r_2, P) = AD(3, r_2, P)$ = ... =
$AD(k, r_k, P) = AD(1, r_k, P) = AD(1, r_1, P)$.

This contradicts with the second term of Guideline 3.1 which states that every routing instance is assigned a globally unique AD value:
$AD(1, r_1, P) \neq AD(2, r_1, P)$,
$AD(2, r_2, P) \neq AD(3, r_2, P)$,
...,
$AD(k, r_k, P) \neq AD(1, r_k, P)$.

To summarize, networks compliant with Guideline 3.1 do not contain any MP dispute wheels. Then, applying the result from Theorem 3.2, we conclude that the route selections converge. □

Although one may expect that network operators will indeed enforce the same AD value across routing processes of the same routing instance, recent empirical studies show that this is not always the case [14].

4.2 Forwarding Loops

The previous section reveals the possible occurence of route oscillations. Guidelines have then been suggested to ensure the convergence of the routes. However, networks compliant with the proposed guidelines can still experience routing instabilities. A network may converge to a stable state that includes a forwarding loop. Section 4.2.1 illustrates such scenarios. Given a configuration, an important question is whether forwarding loops can form. Section 4.2.2 proves this problem to be NP-hard. Because of the complexity, Section 4.2.3 examines the origins of the loops and derives sufficient conditions guaranteeing loop-free forwarding paths.

4.2.1 Illustration of Forwarding Loops

We consider the BGP autonomous system (AS) depicted in Figure 4. The network deploys two routing instances: BGP to learn routes from other ASes and an IGP (e.g., EIGRP 1) to exchange routing information within the AS. Such deployment is typical of BGP networks.

We assume that routers C and D are BGP Route Reflectors (RR) to routers A and B. To avoid a single point of failure, multiple route reflectors are commonly deployed within a same cluster. Both C and D are RRs for the same cluster so that when one fails, the other can take over.

We focus on a destination prefix P received from an external BGP neighbor at routers A and B. Routers C and D receive the routes (through the iBGP network) and we assume that local policies are such that router C prefers A as the egress node whereas router D favors B as the egress point for P.

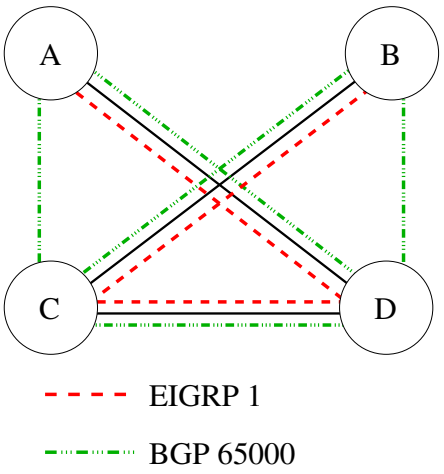

Figure 4: Illustration of a permanent forwarding loop.

We implemented the configuration and observed the presence of a permanent forwarding loop between routers C and D:

- Router D points to router C as the immediate next-hop for prefix P. This is because B is the selected egress point for D, and in order to reach B, D forwards the traffic to C.
- Router C points to router D as the immediate next-hop for prefix P. This comes from the fact that A is the egress point for C, and to reach A, C forwards its traffic to D.

The loop C-D-C forms because routers select a path to an egress point but intermediate nodes on that path adopt a different route to the destination. Such behavior is commonly refered to as a path deflection. [9] is the first study to disclose the possible formation of forwarding loops because of improprer iBGP configurations. [11] identified a set of sufficient conditions guaranteeing loop-free forwarding paths. However, the conditions do not always suffice. The configuration from Figure 4 is indeed compliant with the suggested conditions but still vulnerable to instability. This is because [11] assumed that iBGP sessions do not contain routing policies. In operational environments, iBGP sessions may actually contain routing policies. More importantly, the problem is not limited to BGP nor policy-based routing protocols. The problem occurs because of the interactions between routing processes. To illustrate it, we consider the following scenario. We assume the topology from Figure 2(a) with the three routing instances now being three instances of OSPF (e.g., OSPF 1, OSPF 2 and OSPF 3). OSPF is a shortest path routing protocol, not a policy-based routing protocol, but loops can still form. The routes to P get flooded in each routing instance. Each border router receives two routes to P and

t_1 B chooses D as its next-hop: B prefers the route from $B.3$ since $AD(3, B, P) < AD(2, B, P)$. B adopts the path B-D-A and selects D as its next-hop.

t_2 D selects C as its next-hop: D prefers the route from $D.1$ since $AD(1, D, P) < AD(3, D, P)$. D adopts the path D-C-A and selects C as its next-hop.

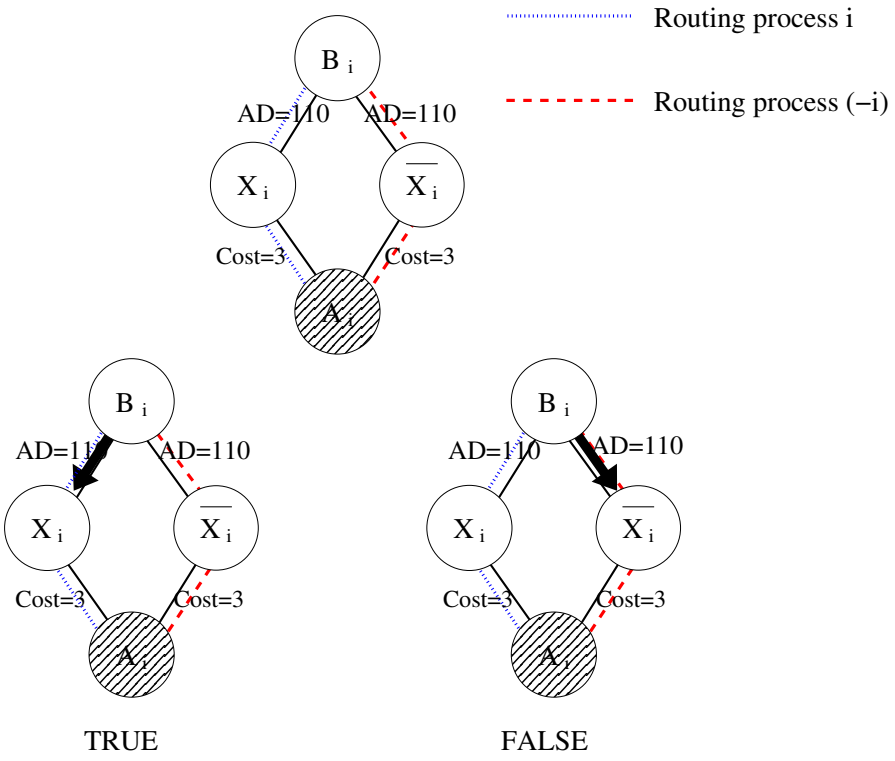

Figure 5: Representation of a variable X_i.

t_3 C selects B as its next-hop: C prefers the route from $C.2$ since $AD(2, C, P) < AD(1, C, P)$. C adopts the path C-B-A and selects B as its next-hop.

Consequently, traffic sent to P can terminate in a forwarding loop. When B receives traffic to P, it forwards it to D, D sends it to C which forwards it back to B.

4.2.2 Complexity of Detecting Loop

Given a configuration, an important question is whether forwarding loops can form. We show that this question can be complex to answer.

Theorem 3.4: *Given a configuration, determining whether the forwarding paths can result in a forwarding loop between the routing instances is NP-hard.*

Proof As the NP-hard proofs in [12] and [15], the proof relies on a reduction from the 3-CNF SAT which is known to be NP-complete.

We consider an instance of 3-CNF SAT, i.e., a set of m clauses of length at most 3 over n Boolean variables $(X_1, X_2, ..., X_n)$: $B = C_1 \wedge C_2 \wedge ... \wedge C_m$. Each clause $C_k, (1 \leq k \leq m)$ is composed of at most three distinct literals: $C_k = l_1^k \vee l_2^k \vee l_3^k$, and each $l_i^k (1 \leq i \leq 3)$ is of the form of X_j or $\overline{X_j}$ $(1 \leq j \leq n)$.

We construct a configuration G such that B is satisfiable if and only if G contains a loop.

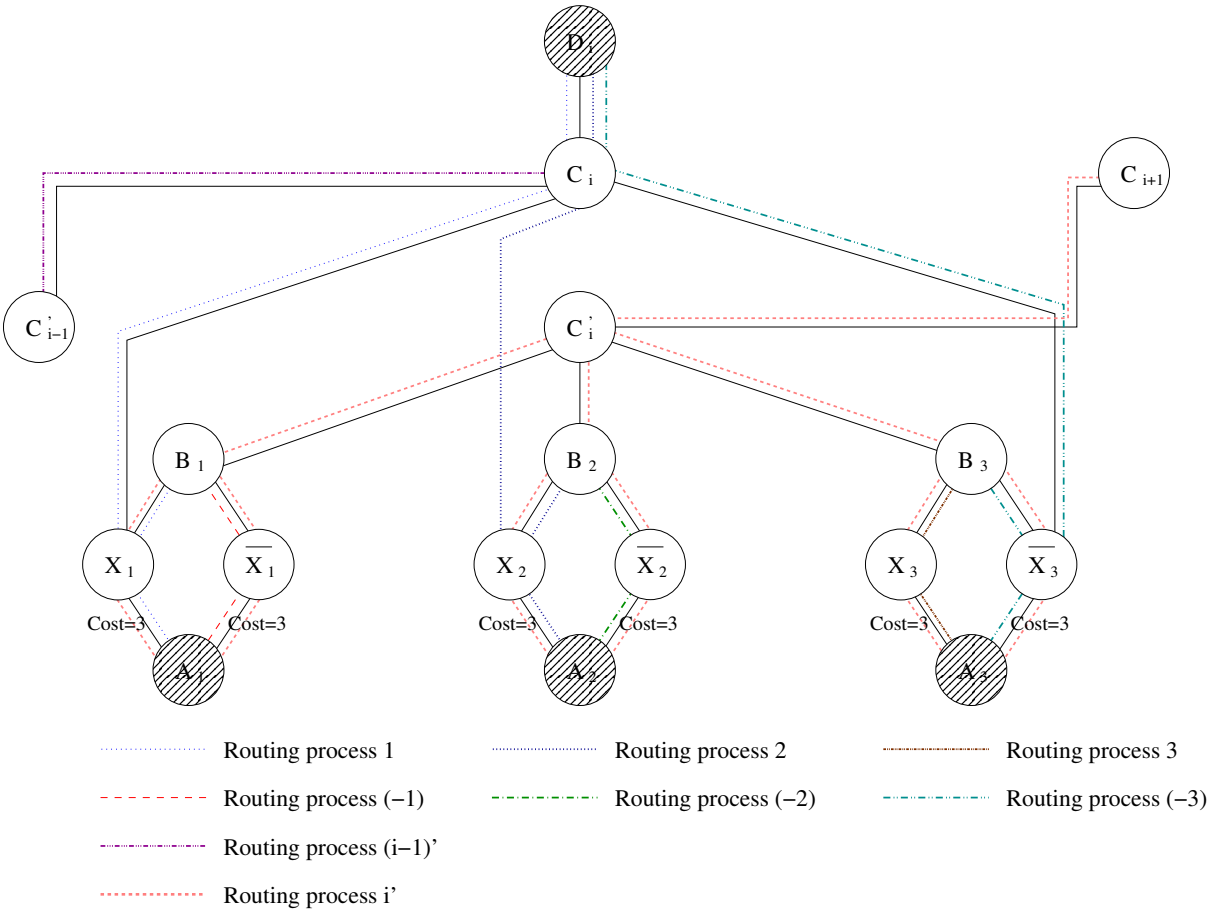

Figure 6: Representation of a clause $C_i = X_1 \vee X_2 \vee \overline{X_3}$.

Each link has a cost of 1 unless specified otherwise. Each variable X_i $(1 \leq i \leq n)$ is represented by the configuration from Figure 5. It consists of four routers A_i, X_i, $\overline{X_i}$ and B_i and two routing instances. Routers A_i, X_i and B_i belong to routing instance i and routers A_i, $\overline{X_i}$ and B_i are part of routing instance $-i$. Router A_i originates a route to P in both i and $-i$. At router B_i, both routing processes have identical AD. As such, B_i selects the route that is received first. When B_i selects the route from i, pointing to X_i as its next-hop, we associate this state with the TRUE value for the variable X_i. Instead, when B_i selects the route from $-i$, pointing to $\overline{X_i}$ as its next-hop, the state is associated the FALSE value for the variable X_i. The links A_i-X_i and A_i-$\overline{X_i}$ have a cost of 3.

For each clause $C_i = l_1^i \vee l_2^i \vee l_3^i$ $(1 \leq i \leq m)$, there exists j_1, j_2 and j_3 in $[1, n]$ such that for every k, $(1 \leq k \leq 3)$, $l_k^i = X_{j_k}$ or $l_k^i = \overline{X_{j_k}}$. As illustrated in Figure 6, for each clause C_i we add three nodes C_i, C_i' and D_i to G such that:

- For each k, $(1 \leq k \leq 3)$, C_i is connected to X_{j_k} if $l_k^i = X_{j_k}$, or to $\overline{X_{j_k}}$ if $l_k^i = \overline{X_{j_k}}$. C_i is also connected to D_i, and the three nodes belong to routing instance j_k (or respectively $-j_k$).
- C_i' is connected to C_{i+1} and to the nodes B_{j_k} which are directly connected to each l_k^i $(1 \leq$

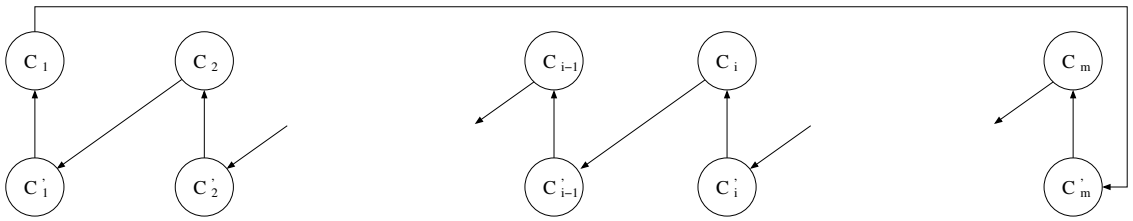

Figure 7: Illustration of the forwarding loop. For all *i modulo m*, $(1 \leq i \leq m)$, C'_i forwards traffic destined to P to C_i (through some intermediate nodes), and C_i sends its traffic to P to C'_{i-1} resulting in a forwarding loop C'_m - C_m - C'_{m-1} - C_{m-1} - ... - C'_1 - C_1 - C'_m - C_m.

$k \leq 3$). These five nodes, X_{j_k} $(1 \leq k \leq 3)$, $\overline{X_{j_k}}$ $(1 \leq k \leq 3)$ and the A_{j_k} which are directly connected to each l^i_k $(1 \leq k \leq 3)$ belong to a distinct routing instance i'. At each router, i' has a higher AD value than those of all the i and $-i$ except at router C_{i+1} where on the opposite, i' has the lowest AD value.

- D_i originates a route to P in all of its routing processes.

The graph G can be computed from B in polynomial times. We now demonstrate that the transformation of B into G is a reduction.

$\Rightarrow$ We show that if B has a satisfying assignment, G contains a forwarding loop between C'_m - C_m - C'_{m-1} - C_{m-1} - ... - C'_1 - C_1 - C'_m - C_m (Figure 7). We demonstrate this result in two steps. First, we prove that for all *i modulo m*, $(1 \leq i \leq m)$, router C'_i forwards some of its traffic destined to P to router C_i. Second, we show that for all *i modulo m*, $(1 \leq i \leq m)$, C_i sends its traffic to destination P to router C'_{i-1}.

Step 1: C'_i belongs to i' and learns three routes with identical minimal cost to P. The egress points are the 3 routers A_j $(1 \leq j \leq n)$ that are directly connected to l^i_k $(1 \leq k \leq 3)$. As such, C'_i load balances traffic destined to P on all three paths. Router C'_i forwards traffic to the routers B_{j_k} $(1 \leq j \leq n)$ that are directly connected to l^i_k $(1 \leq k \leq 3)$. Because B has a satisfying assignment, at least one of the l^i_k $(1 \leq k \leq 3)$ must have the TRUE value. We assume that l^i_h = X_l or $\overline{X_l}$ $(1 \leq h \leq 3)$ $(1 \leq l \leq n)$ is one of them. In such case, the router B_l connected to l^i_h forwards traffic to router X_l if l^i_h = X_l or to $\overline{X_l}$ if l^i_h = $\overline{X_l}$. This latter router may be running multiple routing processes, but routing instance l (or respectively $-l$) has the lowest AD value. This process may be offering multiple routes to P depending on the number of clauses that have X_l (or respectively $-X_l$) in it. The selected egress points are the routers D_j directly connected to the routers C_j such that the corresponding clause C_j includes X_l (or respectively $-X_l$). Because all these routes have equal cost, router l^i_h load balances traffic on all those routes. Part of the traffic is sent to the next-hop C_i. We have thus shown that C'_i sends some of its traffic destined to P to C_i.

Step 2: C_i belongs to multiple routing instances including $(i-1)'$. This later has the lowest AD value and offers a route to P. It will therefore be selected, and because of the topology of the network, C_i forwards its traffic to C'_{i-1}.

$\Leftarrow$ We show that if B has no satisfying assignment, G contains no forwarding loop. We assume that B has no satisfying assignment. In such case, there exists a clause C_i, $(1 \leq i \leq m)$, that has

the FALSE value. Considering the corresponding routers C_i and C'_i, C'_i forwards none of its traffic to C_i. Starting from any node in G, we can verify that the traffic reaches an originating node and does not include any forwarding loop. □

4.2.3 Analysis of Root Cause

Given the complexity of the problem, we analyze the root causes of the problem and we derive a number of sufficient conditions that guarantee loop-free forwarding paths.

To identify the origins of the forwarding loops, we first model the interactions between the routing processes and more specifically the route selection at each router.

We consider a network with m routing instances (1, 2, ..., m). For a router r, we note $C(r)$ the set of subnets that are directly connected to r. We focus on a destination prefix P. Router r may be running multiple routing processes. The function $b(i, r, P)$ gives the next-hop of the route learnt from routing process i at router r to prefix P. For example, we may have $b(i, r, P) =$ "$192.168.1.1$". If routing instance i does not have a route to P at r, then $b(i, r, P) = \emptyset$.

If several routing processes offer routes to the same destination prefix P, the routing process with the lowest AD is selected at r. As mentionned in Section 2, it is called the *selected routing process*. The selected routing process $s(r, P)$ – ($s(r, P) \in \{1, 2, ..., m\}$) – for prefix P at router r is defined as:

$s(r, P) = argmin_k \{AD(k, r, P) \mid b(k, r, P) \neq \emptyset\}$.

As such, $b(s(r, P), r, P)$ represents the address of the next-hop of the selected route to P at router r. Because of different reasons (e.g., IP tunnels, static routes, BGP, etc.), $b(s(r, P), r, P)$ may point to an address that is not directly reachable by r. For example, packets may be tunneled to a firewall that is multiple hops away. The logical link needs to be mapped to a physical path. The immediate next-hop where traffic to P is forwarded to, at router r is determined by the following recursive function:

$nxt(r, P) =$

1: **if** $b(s(r, P), r, P) \in C(r)$ **then**
2: return $b(s(r, P), r, P)$
3: **else**
4: return $nxt(r, b(s(r, P), r, P))$
5: **end if**

It is important to note that when multiple routes from the selected routing process have equal minimum cost, the router load balances the traffic to these minimum equal cost routes. As such, the output of $b(s(r, P), r, P)$ can be a set of IP addresses.

This representation allows us to formally define a path and to characterize a deflection.

Definition 3.1 Given a router r and a destination prefix P, a *path* from r to P is a sequence of nodes $ra_1...a_k$, such that for each i ($1 \leq i \leq k$), $a_i \in nxt(a_{i-1}, P)$ and either

1) $P \in C(a_k)$ or,

2) $nxt(a_k, P) = \emptyset$ and for each i ($i < k$), $nxt(a_i, P) \neq \emptyset$.

We say that a path includes a deflection when it is composed of segments learnt from different routing instances. We distinguish two types of deflections: the intra-router route deflection and the inter-router route deflection.

Definition 3.2 An intra-router route deflection occurs when there exists a router r and a prefix P such that the iterations in $nxt(r, P)$ involve different selected routing processes.

Definition 3.3 An inter-router route deflection happens at router r with respect to P if there exists another router r' such that $r \in b(s(r', P), r', P)$, $s(r, P) \neq \emptyset$ and $s(r', P) \neq s(r, P)$.

Deflections are responsible for the observed forwarding loops. In Figure 4, the network includes two intra-router route deflections at routers C and D. To reach destination prefix P from router C, the first iteration of $nxt(C, P)$ points to A with iBGP being the selected routing process. Because A is not directly reachable from C, a second iteration of $nxt()$ is required. The second iteration of $nxt(C, P)$ points to D with OSPF being the selected routing process. The resolution of the immediate next hop to P involves two iterations for $nxt()$ with differing selected routing processes. Similarly, an intra-router route deflection occurs at router D resulting in the forwarding loop between C and D. In the other scenario from Section 4.2.1 (topology from Figure 2 with the three routing processes being three instances of OSPF), the loop forms because of the occurrence of inter-router route deflections at routers B, D and C. We show that the presence of route deflection is a necessary condition for forwarding loops and therefore, the absence of route deflection is a strong sufficient condition for loop-free forwarding paths.

Theorem 3.5: *That the network contains a route deflection is a necessary condition for forwarding loops.*

Proof We prove it by contradiction. We consider a network composed of multiple routing instances. We assume that the interactions between the routing instances converge to a state containing a forwarding loop and that the forwarding paths are devoid of route deflections. The absence of deflections implies that, within and across routers, all the next hops composing the paths to the destination are learnt from a single routing instance. Because each routing process is assumed to be correct, i.e., to converge to a loop-free state, the forwarding paths are devoid of loops. This is in contradition with the first assumption. □

Based on Theorem 3.5, we derive the following sufficient condition: for a network that has converged, the absence of route deflections is a sufficient condition guaranteeing loop-free forwarding paths. We then propose the following configuration guideline.

Guideline 3.2 *Originate each prefix in a single routing instance.*

Theorem 3.6: *Guideline 3.2 guarantees the absence of inter-router route deflections.*

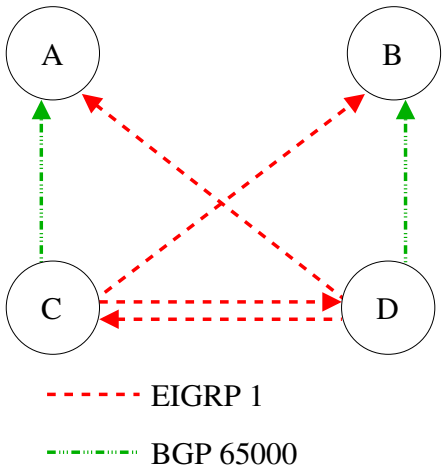

Figure 8: Illustration of a cycle (C-D-C) in the mapping of the forwarding paths of the network in Figure 4.

Proof We prove it by contradiction. We assume that each prefix P is originated in a single routing instance and we assume that there exists two distinct routers r and r' such that $b(s(r', P), r', P) = r$, $s(r, P) \neq \emptyset$ and $s(r', P) \neq s(r, P)$. The first term, $b(s(r', P), r', P) = r$, implies that P is advertised within routing instance $s(r', P)$. The second term, $s(r, P) \neq \emptyset$ implies that P is also advertised within routing instance $s(r, P)$. The final term, $s(r', P) \neq s(r, P)$, means that P is advertised in two distinct routing instances. This is in contradiction with the first assumption that each prefix P is originated in a single routing instance. □

Guideline 3.2 recommends to originate a prefix in a single routing instance. To propagate a destination prefix into other routing instances, an operator can make use of route redistribution. [13] identified guidelines for a safe and robust route redistribution. The next section further analyzes the interactions between route selection and route redistribution.

Guideline 3.2 allows an operator to configure a network in such a way that the network is devoid of inter-router route deflections. However, the second category of deflection, the intra-router route ones, can be more difficult to suppress. This is because overlay routing protocols (e.g., BGP) and routing features (e.g., OSPF forwarding address) rely on these intra-router route deflections. Eliminating intra-router route deflections may be too restrictive and prevent operators from achieving their objectives. As such, we propose to relax the previously identified sufficient condition.

Section 3 pointed out that the forwarding paths from one routing instance to a destination P can be represented by a directed acyclic graph. We note that the presence of a forwarding loop implies a cycle in the mapping and the union of the directed acyclic graphs from the different routing instances. We highlight the cycle in the previous configurations to clarify what we mean by the mapping and the union of the directed acyclic graphs. Figure 8 represents the forwarding paths to P from the network in Figure 4. Routers A and B have a route to P. C and D learn the routes through iBGP and C points to A as the next hop while D points to B. As such, we have two edges

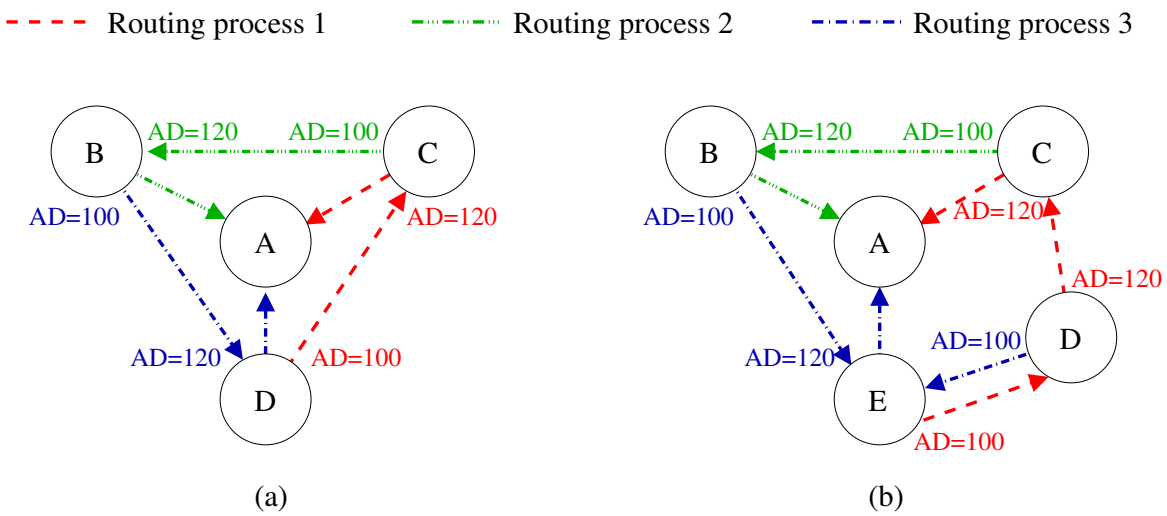

Figure 9: Illustration of cycle(s) in the union of the forwarding paths in the second scenario from Section 4.2.1. The cycle(s) consist of B-D-C-B in (a) and D-E-D, B-E-D-C-B in (b).

$<C, A>$ and $<D, B>$. Because A is not directly reachable from C, and B is not from D, we map each of these logical links to a phyical path. For C to reach A, the forwarding paths from EIGRP 1 indicate C-D-A. Similarly, for D to reach A, the forwarding paths from EIGRP 1 consist of D-C-A. The union of the forwarding paths reveals a cycle C-D-C. Similarly, Figure 9 discloses the presence of cycles in the union of the forwarding paths in the networks from the other scenario from Section 4.2.1. From these observations, we can derive a weaker sufficient condition allowing network operators to verify that the forwarding paths are loop-free.

Theorem 3.7: *For a network that converges, the absence of cycle in the mapping and the union of directed acyclic graphs guarantees that the forwarding paths are devoid of loops.*

Proof We assume the presence of a forwarding loop in the network. More specifically, we assume that there exists a destination prefix P such that the forwarding paths result in a loop. Because datagrams flow along the forwarding paths, the presence of the forwarding loop implies the existence of a cycle in the mapping and union of the directed acyclic graphs corresponding to P. □

5 Interplay between Route Selection And Redistribution

The previous section looked at routing anomalies that can derive from route selection. This section analyzes new instabilities that can appear when considering the additional usage of route redistribution on top of route selection. The additional usage of route redistribution can create a new range of instabilities which are not limited to loops and oscillations but also include non deterministic forwarding paths.

For loops and oscillations, it is important to note that the origins of the problems when route

selection and route redistribution are employed simultaneously differ from those when route selection is used solely. These differences are important because consequently the absence of MP dispute wheels and route deflections – main causes of instabilities in route selection – no longer guarantees instability-free routing. [15] looked at routing anomalies that can derive from route redistribution. Section 5.1 summarizes the results from [15] focusing on the origins of the anomalies.

Then, Section 5.2 adds to that work by disclosing additional forms of routing instabilities that can result from the interplay between route selection and route redistribution. Reports have described a number of unacceptable behaviors when deploying route redistribution. None of the existing models can explicate the observed outcomes. We argue that existing models are too restrictive and do not properly represent the intricate dependencies between route selection and route redistribution.

Consequently, Section 5.3 proposes a more comprehensive functional model that allows the analysis and the understanding of the interactions between these two procedures.

5.1 Summary of Related Work on Routing Loops And Oscillations

When deploying route redistribution, [15] analyzed routing anomalies that can derive from it. The study revealed that instabilities can form because of two main factors: *MP dispute wheels* and *history-less routes*.

[15] showed that route redistribution behaves like a vector protocol. When a routing instance i announces a route to destination P into instance j, j does not have a global view of the topology but only knows that i is the "next-hop" for P. Combined with the concept of administrative distances which allows a routing instance to prefer routes from a neighboring instance independently of other attributes, route redistribution resembles BGP and the LOCAL PREFERENCE attribute. As such, conflicting routing policies can result in MP dispute wheels which similar to the dispute wheels in BGP [12] can cause route oscillations. A notable difference with the results from the previous sections is that with route redistribution, MP dispute wheels involving routing instances that are link-state can result in route oscillations.

The second source of the problems is the lack of *history* in the redistributed routes. Whenever a route is redistributed from a source routing process into a target routing process, all the attributes of the routes are generally reset to arbitrary values. This is in part due to the incompatibility of metrics between routing protocols (e.g., RIP metrics versus EIGRP metrics). However, the history of a route is an essential element to suppress routing instabilities. For example, the hop count in RIP and the AS-PATH in BGP prevents a router from selecting a route that it formerly advertised. In the absence of such information, a router can select any of the redistributed routes, potentially resulting in forwarding loops, route oscillations or sub optimal routing.

[15] and [13] identified guidelines supporting not only safety and validity but also additional desired properties for route redistribution such as robustness and domain backup.

5.2 Nondeterministic Routing Behaviors

Route redistribution can cause additional problems to the routing anomalies discussed so far. Companies have reported scenarios that can produce unexpected and non-deterministic forwarding

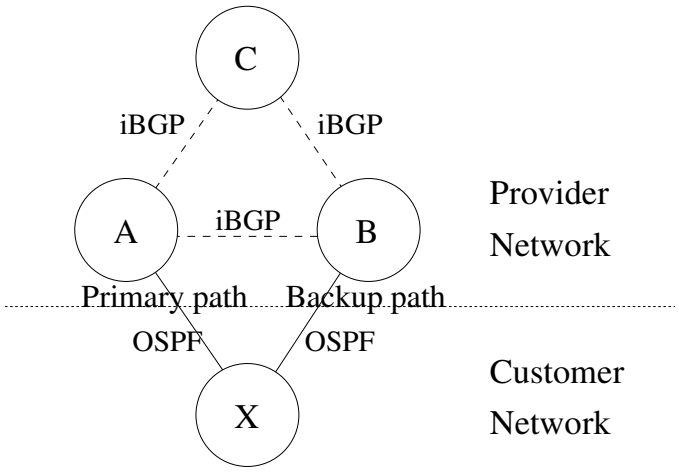

Figure 10: Non-deterministic forwarding paths.

paths [3]. Several models [7], [15], [17] have been developed to study the interactions between routing processes with varying levels of granularity. Yet, none of them can explicate the observed forwarding paths.

Section 5.2.1 illustrates the problem. Section 5.2.2 examines the extensiveness of the problem. We show that the behaviors can be observed with different protocols including BGP, RIP and OSPF. Section 5.2.3 analyzes the causes of the oberved behaviors. We argue that prior work could not justify the problem because treating each procedure – route selection and route redistribution – separately is not sufficient. Instead, it is the interplay between them that is responsible for the observed behaviors.

When a router R redistributes a route from a routing process u into a routing process v, how should the locally redistributed route be treated in v? Should the locally redistributed route be considered for local route selection? Should a locally redistributed route be considered in routing process v's best path selection algorithm? These are important questions that directly impact the selection of the active route. However, there is no framework to analyze these questions.

5.2.1 Motivating Scenario

The scenario is first described in [3]. We consider the network depicted in Figure 10. It consists of a provider network offering Internet service to a customer network through two paths: (A-X) and (B-X). The routers (A, B, C) in the provider network run an IGP and form a full iBGP mesh. The provider network learns from the customer's routes through static routes at routers A and B. At A and B, the static routes, pointing to the customer's network, are redistributed into BGP so that they can be further propagated to other BGP networks. We assume that the customer wants to use the A-X link as the primary one for traffic arriving from the service provider, and B-X is used as a backup. As such, at router B, the BGP process is configured with a lower AD value than that of the static routes. When the A-X link is up, B should forward the traffic to the customer via A.

B should receive two routes to X: the first one from router A through iBGP and the second one being the static route. Because of the configuration at B, the first route should always be prefered.

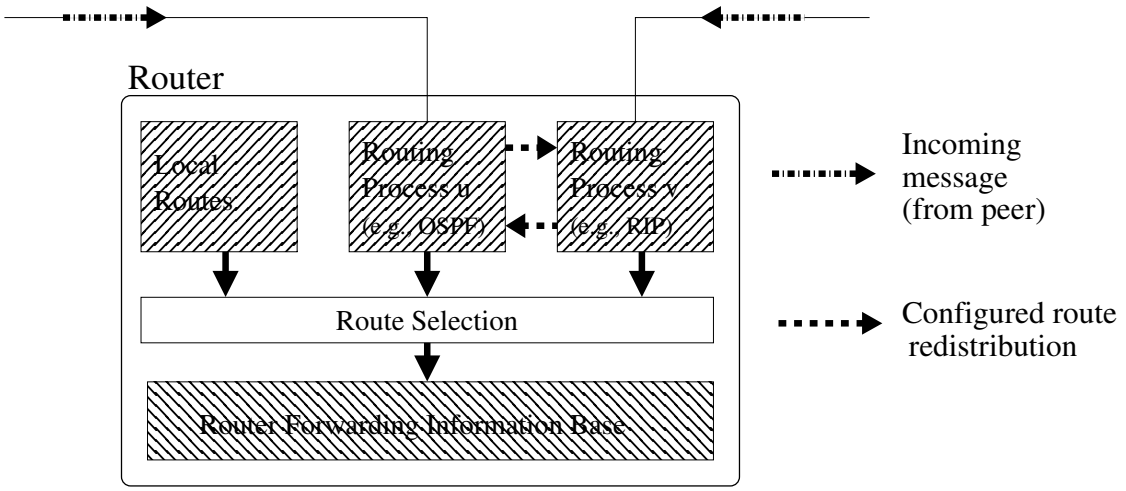

Figure 11: Experiment. A router is configured with two routing processes (u, v) and receives two routes to the same prefix. Mutual route redistribution is configured between u and v.

However, [3] reported that the forwarding paths at B surprisingly depend on the order of the message arrival. Such behavior is clearly unacceptable for network designs. We have implemented the topology using 4 Cisco 3600 routers IOS Version 12.2 and we observed the following behaviors at router B:

- *Case 1* When the iBGP route from router A is the first considered route, it becomes the active route. Then, when the static route is considered, the iBGP route remains the active one because of its lower AD value. This is indeed the intended result.
- *Case 2* When the static route is considered 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 OSPF route, the static route remains the active route. The route with the highest AD value is unexpectedly the active route. Contrary to the design goals, B forwards traffic to the customer directly to X and announces such route to other BGP neighbors. The link B-X is not used as a backup path.

5.2.2 Extensiveness of Problem

This section examines the extent of the previously observed problems. We seek to understand whether the behaviors are specific to one implementation or e.g., to one routing protocol. Such understanding helps to identify the root causes of the problem (e.g., incorrect implementation, incomplete specifications, conflicting procedures). To answer these questions, we conduct the following experiments.

We consider three implementations: Cisco 3600 IOS version 12.2(24a), Quagga Software Routing Suite version 0.98.6 [1] and XORP version 1.4 [2].

Source of Routes		Implementation		
Primary	**Backup**	**Cisco**	**Quagga**	**XORP**
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 results. The symbol "✓" indicates that independently of the message arrival order, the route with the lowest AD value is indeed selected as the active route, i.e., the outcomes are as expected. The symbol "✗" signifies that the message arrival order impacts the active route. Depending on the arrival order, the route with the highest AD can become the one used for forwarding purposes.

For each implementation, we configure a router with two routing processes u and v. One of the routing processes is configured with a lower AD value to become the *Primary* path. The other routing process should only serve as a *Backup*. Route redistribution is configured from u to v and vice-versa. Then, we advertise two routes to the same destination prefix P (one in u, and another in v) from neighboring routers (Figure 11). We analyze whether the order of the injected messages impacts the outcome of the route selection and route redistribution procedures.

As explained in Section 2, when two routes are present, the route with the lowest AD value should become the active route and the one that is redistributed. The results of the experiments are summarized in Table 1. The symbol "✓" indicates that independently of the message arrival order, the route with the lowest AD value is indeed selected as the active route, i.e., the outcomes are as expected. Instead, the symbol "✗" signifies that the message arrival order impacts the active route. Depending on the arrival order, the route with the highest AD can become the one used for forwarding purposes.

We note the following observations. First, all implementations can produce unexpected outcomes. Each implementation ends up selecting a route with a higher AD value as the active route for some configuration. The problem is therefore pervasive. Second, we observe inconsistencies across the implementations: each router can generate a different outcome given the same set of inputs. These results suggest that parts of the problem are due to incorrect implementations. We argue that these problems are beyond implementation errors but come from a lack of model to understand, reason and support the interactions between route selection and route redistribution. The next section actually shows that existing documents can instead be misleading and be responsible for those erroneous implementations.

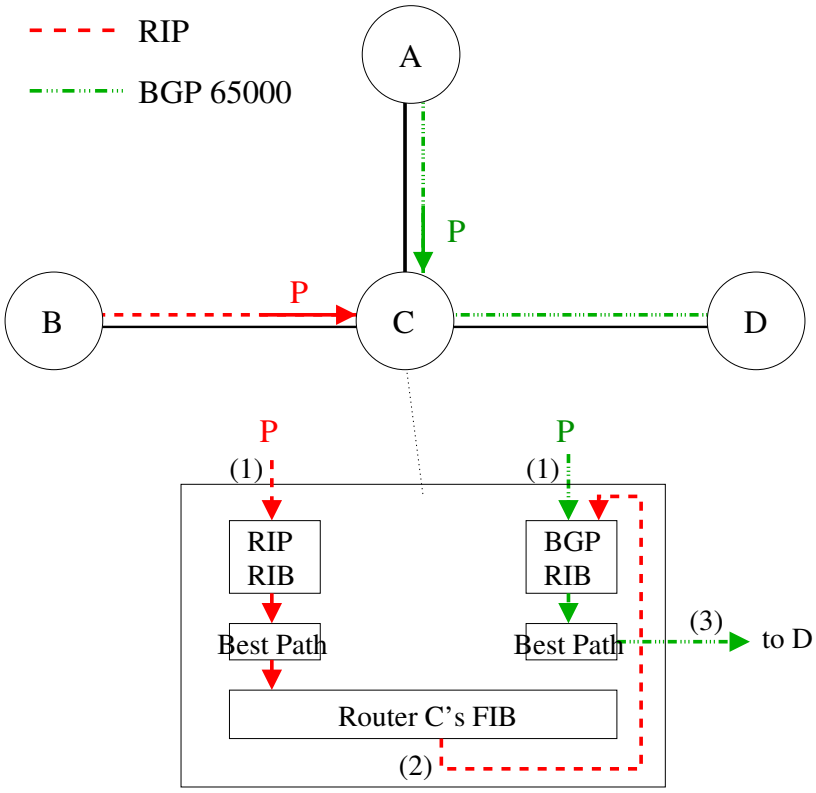

Figure 12: Illustration of additional route deflections in existing implementations.

5.2.3 Analysis of Root Cause

It is difficult to pinpoint the root causes of the observed behaviors because of the inaccessibility to the source code of the commercial implementations and the scarce documentation on this topic.

As suggested by [3], a look at the RIBs shows that the locally redistributed routes are typically stored in the target routing process' RIB. When the static route is locally redistributed into BGP (Section 5.2.1, Case 2), the locally redistributed route is present in the BGP RIB. In fact, by default, the locally redistributed routes present a higher preference than any other incoming iBGP route because of a higher default WEIGHT value for locally redistributed routes. The WEIGHT is a Cisco-specific attribute [4] and is the first considered parameter in the BGP best path selection of Cisco routers. Routes with a higher WEIGHT are prefered. Consequently, the execution of the BGP best path selection algorithm selects the locally redistributed route as its best path. BGP routes from neighbors are not presented to the inter routing processes route selection algorithm. Then, for stability reasons, the locally redistributed route is filtered out and not presented to the inter routing processes route selection algorithm either [15]. This explicates the reasons the iBGP routes from the BGP neighbors cannot become the active route despite a lower AD value.

In fact, in addition to producing non-deterministic forwarding paths, such architecture can cause further routing anomalies. We consider the network from Figure 12.

(1) Router C receives two routes to the same prefix P from a RIP peer and BGP neighbor. We

assume that the route from RIP becomes the active route because of a lower configured AD value.

(2) The active route from RIP is redistributed into BGP. The BGP RIB contains two routes to P and the route from the BGP neighbor may be prefered by the BGP best path selection algorithm to the locally redistributed route (e.g., because of a route-map setting a high WEIGHT value to routes received from BGP neighbors).

(3) As such, router C advertises the BGP route received from its BGP neighbor (A) to other BGP neighbors (D) instead of the locally redistributed route.

Router C advertises a route that is different than its active route. A router advertising a route that is not active can cause unexpected inter router route deflections, which may further result in sub optimal routing, policy violations and forwarding loops as described in Section 4.2.1.

These observations are only part of the problems. They do not explain all the observed results. We discovered that each implementation may adopt a different architecture. The study of the Quagga source code indeed revealed that when a route is locally redistributed into the RIP protocol, all RIP messages received from the neighbors are in fact discarded independently from the AD values. This explicates the observed outcomes with the RIP protocol when using the Quagga implementation.

We postulate that the lack of precise specifications of route selection, route redistribution and the ways these two processes should interact leads to violation of the route selection property (P1, Section 2). In the next section, we propose a functional model for the interactions between routing processes that guarantees both route selection and route redistribution properties (Section 2).

5.3 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 that makes the dependencies between route selection and route redistribution procedures unambiguous and guarantees both the route selection and route redistribution properties (Section 2).

Section 5.3.1 describes a potential solution for vector protocols. Then, Section 5.3.2 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 5.3.3 shows that the proposed functional model guarantees the two properties given in Section 2.

5.3.1 A Functional Model for Vector Protocols

The proposed solution for vector protocols is depicted in Figure 13. Each vector routing process (e.g., RIP or EIGRP) has two RIBs: *RIBin* for incoming route announcements and *RIBout* for outgoing advertisements. A new announcement from a peer must first through some *filters*. The

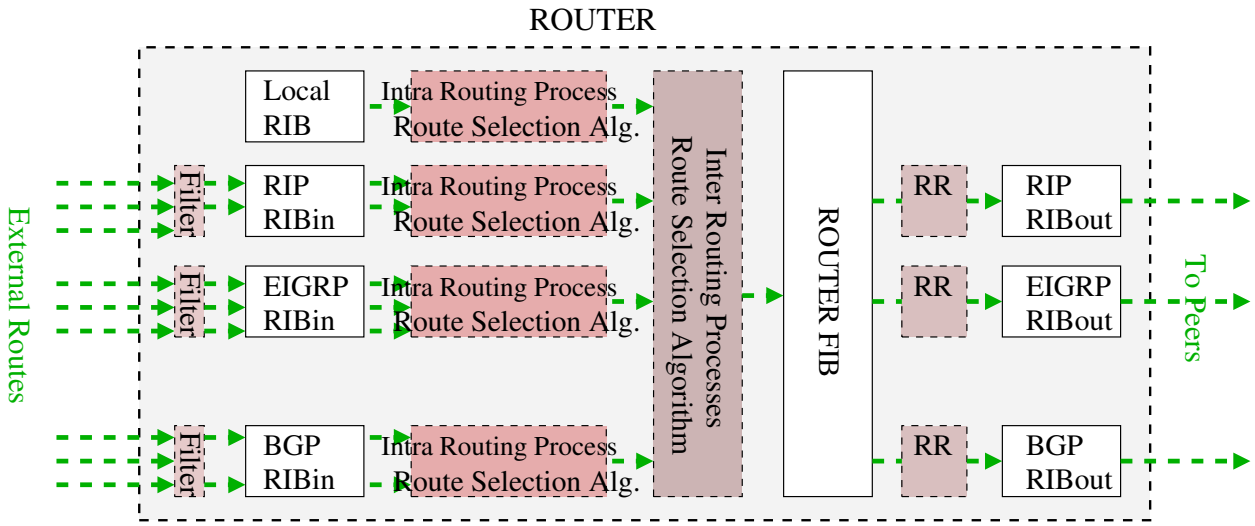

Figure 13: A functional model for vector protocols.

filters discard invalid advertisements and routes not compliant with local policies. For RIP, routes whose metric exceeds 16 are filtered. Similarly, a BGP advertisement whose AS_PATH includes the local AS number is dismissed. All routes are stored in the *RIBin* after passing the filters. A *protocol specific route determination algorithm* determines the most preferred route among all routes to the same prefix.

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 original process before advertised out. 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 each time the active route is redistributed.

In this model, a locally redistributed route will never be 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.

5.3.2 Extension for Link State Protocols

This section extends the vector model to accommodate link state protocols. As depicted in Figure 14, 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 RIBs. EIB is used to track the routes that are eligible to become active routes at the router. Built-in filters (i.e., *Filters'* in Figure 14) between RIB and EIB are used to discard locally redistributed routes from the RIB. 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

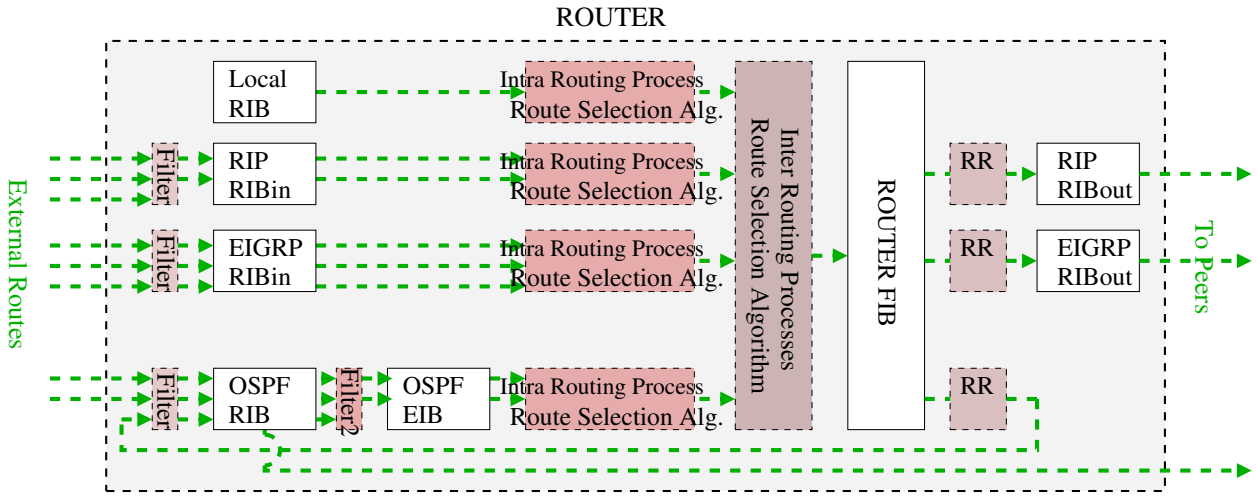

Figure 14: A functional model supporting all protocols.

from the perspective of the route selection procedure.

Active routes can optionally be redistributed into other routing processes. If the target routing process is a vector protocol, the redistributed route is added to the target routing process' RIBout. If the target routing process is a link state protocol, the redistributed route is inserted into the target routing process' RIB. Similarly, when a route is redistributed from a vector routing process into a link state routing process, the redistributed route is injected into the target routing process' RIB.

5.3.3 Correctness of the Proposed Model

The following theorem establishes the correctness of our model.

Theorem 3.8: *The proposed model guarantees route selection property P1 and route redistribution property P2 that are presented in Section 2.*

Proof sketch. Consider a router and a destination prefix P. Let $S(P)$ denote the set of of routes to P that either come from a neighbor or are static routes local to the router. First, the local policies are applied to the external routes of $S(P)$ and non compliant routes are filtered out. Each remaining valid route is stored in either a RIB or a RIBin. Second, contrary to the existing implementations, locally redistributed routes are excluded from consideration by any of the protocol specific route determination algorithms. As such, each routing process that receives an external route presents a non empty set of routes to the route selection procedure. This eliminates the error condition as described in Section 5.2.3. Therefore, the model guarantees property P1. Furthermore, in this model, a route is redistributed directly from the router's FIB. Therefore, the model guarantees P2, i.e., a route can only be redistributed if active. □

6 Related Work

[7] mentions that route selection can cause forwarding loops but does not provide any illustration nor guideline to avoid them. Instabilities due to route redistribution are documented in [6], [5], and [13] and [15] presents a model for analyzing such instabilities. Our work is the first to illustrate how routing instabilities may result from route selection alone or its interplay with route redistribution. We also analyze the root causes of these instabilities and develop guidelines or solutions for avoiding them.

Several studies looked at the interactions between BGP and its underlying IGP. [19] revealed that such interactions can delay the convergence of BGP. [11] disclosed instabilities that may result from certain iBGP configurations. It introduced a model to analyze the instabilities, proved important results regarding the complexity of detecting such problems, and proposed sufficient conditions for guaranteeing the correctness of iBGP configurations. However, we show in Section 4.2 that the sufficient conditions identified in [11] do not always suffice. [10] also analyzed routing anomalies caused by the interactions between BGP and its underlying IGP. It proposed a taxonomy of desirable properties for routing protocols, presented a general framework to study the compliance of routing protocols (particularly BGP) with these properties. In comparison to these studies, the scope of our work is much broader. 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 forwarding loops.

7 Conclusion and future work

We have presented a comprehensive analysis of the route selection and route distribution procedures to characterize their vulnerability to different classes of routing instabilities. The results suggest a twofold conclusion. On the one hand, the news is somewhat bleak. These procedures are highly susceptible to routing anomalies and the range of anomalies is much wider than previously reported. 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 a similar attention from the networking community.

To move forward, it is essential to determine if there is a fundamental trade-off between functionality and safety when interconnecting routing protocols. If the two requirements cannot be reconciled, extensions to the current routing selection and route redistribution procedures may need to be developed. A better understanding of *operational requirements* for the interactions between routing protocols is crucial to such an endeavor. This may be achieved by examining the configurations of existing operational networks. The ultimate goal is to derive guidelines that not only ensure the safety of the configurations but also allow operators to achieve their objectives.

References

[1] Quagga Software Routing Suite. http://www.quagga.net/.

[2] XORP: eXtensible Open Router Platform. http://www.xorp.org.

[3] Enke Chen and Jenny Yuan. Deterministic Route Redistribution into BGP.

[4] Cisco. BGP Best Path Selection Algorithm, 2006. Available at http://www.cisco.com/warp/public/459/25.shtml.

[5] Cisco. OSPF Redistribution Among Different OSPF Processes, 2006. Available at http://www.cisco.com/warp/public/104/ospfprocesses.pdf.

[6] Cisco. Redistributing Routing Protocols, 2006. Available at http://www.cisco.com/warp/public/105/redist.pdf.

[7] Cisco. Route Selection in Cisco Routers, 2006. Available at http://www.cisco.com/warp/public/105/21.pdf.

[8] Cisco. What is Administrative Distance?, 2006. Available at http://www.cisco.com/warp/public/105/admin_distance.html.

[9] Rohit Dube. A Comparison of Scaling Techniques for BGP. In *Proc. Computer Communication Review, Volume 29, Number 3*, 1999.

[10] N. Feamster and H. Balakrishnan. Towards a Logic for Wide-Area Internet Routing. In *Proc. ACM SIGCOMM Workshop on Future Directions in Network Architecture*, 2003.

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

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

[13] Franck Le and Geoffrey Xie. On Guidelines for Safe Route Redistributions. In *Proc. ACM SIGCOMM Workshop on Internet Network Management (INM)*, 2007.

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

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

[16] Ratul Mahajan, David Wetherall, and Thomas Anderson. Mutually Controlled Routing with Independent ISPs. In *Proc. NSDI*, 2007.

[17] 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.

[18] John W. Stewart. *BGP4, Inter-Domain Routing in the Internet*. Addison-Wesley, 2001.

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

[20] Russ White, Don Slice, and Alvaro Retana. *Optimal Routing Design*. Cisco Press, 2005.