2011-12

On Route Aggregation

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

On Route Aggregation

Franck Le
IBM T. J. Watson
fle@us.ibm.com

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

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

ABSTRACT

Route Aggregation (RA), the method to supersede a set of routes by a single, more general route, is a fundamental mechanism to the Internet scalability. Yet, despite its importance, it is poorly understood. We present the first systematic analysis of RA via both bottom-up experimental and top-down analytical approaches. We first conduct a set of experiments on RA behaviors of all major routing protocols as implemented by the two leading router vendors. Our experiments show that the RA behaviors vary significantly across routing protocols and vendors. We propose two router level primitives and incorporate them into a canonical router model. The new model captures the diversity of the observed behaviors. With aid of the model, we have advanced the fundamental understanding of RA on three fronts. First, we expose four new types of routing anomaly that can derive from RA. Configuring RA on one router interface can influence how routes are advertised on other interfaces of the same router, impacting network reachability in surprising ways. Second, we demonstrate that determining whether a RA configuration can result in persistent forwarding loops is NP-complete. Finally, we present sufficient conditions for RA primitives to guarantee routing safety, and explore clean-slate designs for RA.

1. INTRODUCTION

Routing scalability is a major concern of the Internet routing system [25]. Route aggregation has played a critical role towards containing this problem so far. Also known as route summarization, route aggregation is a router mechanism that generates a summary route from a set of child routes falling under a common parent prefix and advertises the single summary route in lieu of announcing all the child routes. It allows routers to handle fewer prefixes and has been crucial in curbing the routing table size – from 1994 to 2009, the Internet has grown 300 times from millions of hosts to hundreds of millions of hosts [4], while the Internet core routing table size has only increased 15 times from 20,000 to 300,000 entries [17, 1].

Despite its importance, route aggregation is poorly understood. As will be discussed in Section 2, misconfiguration and mis-use of route aggregation can lead to serious routing anomalies such as persistent packet forwarding loop and blackholes. Due to commercial and operational considerations, ISPs usually do not disclose the details of routing anomaly incidents nor the router configuration files that are needed to shed light on these incidents. This makes it difficult to conduct a white-box analysis study to assess how prevalent these routing anomalies are in practice and to which extent route aggregation contributes to them. As a consequence, researchers resort to blackbox measurement methodologies to study routing anomalies. While this methodology can expose the anomalies in the wild, it is difficult to identify the root cause of the anomalies observed. For example, Xia *et al.* discovered a large number of persistent forwarding loops in the Internet and conjectured that misconfiguration with respect to route-aggregation capabilities could be the root cause behind 50% of them [29]. Another study reported that routing blackholes are prevalent in the Internet but did not give specific reasons [19].

In this paper, we take the position that it is important to understand the risk that route aggregation poses to the correct operation of routing systems. Given the practical difficulties of a white-box analysis and limitations of the black-box measurement methodology, we adopt the following methodology: We first study the route aggregation behavior as implemented in today's routers. The reason for this starting point is that there has been no precise specification of the route aggregation behaviors — RFC 1338 [14], which introduced the concept of route aggregation, only gave a high level goal and description. Many important questions related to route aggregation are left unanswered in RFCs. For example, when should routers aggregate route information? How is the aggregate prefix determined? On which interfaces should an aggregate be advertised? How is the metric of an aggregate route decided? Subsequent RFCs [13, 24] did not clarify. In practice, the exact behaviors of route

aggregation are determined by the specific implementations of route aggregation for different routing protocols and by different vendors. As a second step, we propose a model that allows us to analyze and reason about route aggregation, identify routing abnormalities that can be root-caused to route aggregation, and design and prove more formal and comprehensive guidelines to ensure safe routing in the presence of routing aggregation.

There are obvious limitations with our methodology: It does not yield all the possible routing anomalies that may occur because of route aggregation. Neither does it prove that the identified anomalies are happening in the real world. Despite these, our research based on this methodology has yielded important insights for understanding route aggregation, studying routing anomalies, designing better practice and guidelines for routing safety, and developing next generation routing protocols. In particular, we have made the following contributions:

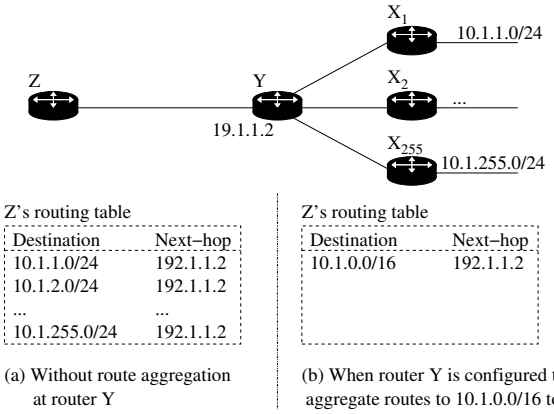
1. We conducted a set of experiments on route aggregation (RA) behaviors of all major routing protocols (BGP, OSPF, EIRGP, RIP) as implemented by the two leading router vendors (Cisco and Juniper). Our experiments show that the RA behaviors vary significantly across routing protocols and router vendors even for simple network setups.

2. We propose two router level primitives and incorporate them into a canonical router model. The new model captures the diversity of observed RA behaviors as implemented by different vendors for different protocols.

3. With aid of the model, we have advanced the fundamental understanding of RA on three fronts. First, we expose four new types of routing anomaly that can result from RA, including permanent route oscillation and unexpected route loss. Furthermore, we identify the root causes for each anomaly (new or previously known). Second, we explain why the current vendor guidelines fall short in addressing the anomalies. In addition, we establish that determining whether a network configuration with route aggregations can result in persistent forwarding loops is NP-complete. Assuming that P is not equal to NP, the problem is untractable. Therefore, we finally present sufficient conditions for the RA primitives to guarantee routing safety. The conditions are independent of routing protocols and work for both Cisco and Juniper designs.

4. We explore and discuss clean-slate designs for RA. We introduce the notion of *negative routes* to generalize the concept of routes, and show how this new concept can safely reduce the number of routing entries.

The rest of this paper is structured as follows. Section 2 provides a brief overview of route aggregation, including two known routing anomalies. Section 3 describes the experiments we performed and the diversity of RA behaviors we observed from the experiments. In Section 4, we propose two router primitives that captures the observed RA behaviors and a canonical router model for analyzing the behaviors. Section 5 focuses on new anomalies and an analysis of their root causes. Then, in Sections 6 and 7, we explain the limitations of current solutions and formulate new safety conditions, respectively. Section 8 explores clean-slate designs for RA. Section 9 briefly discusses related work, and the paper concludes with Section 10.



(a) Without route aggregation at router Y

(b) When router Y is configured to aggregate routes to 10.1.0.0/16 to Z

Figure 1: Route aggregation allows router Y to combine multiple routes (10.1.1.0/24, ..., 10.1.255.0/24) into a single one (10.1.0.0/16).

2. BACKGROUND

Route aggregation (also commonly called *route summarization* or *supernetting*) designates the method to supersede a set of routes by a single more general route. To illustrate this feature, we consider the network depicted in Figure 1. We assume all routers to run a common routing protocol (e.g., RIP or EIGRP). Every router X_i ($1 \leq i \leq 255$) is directly connected to an interface with IP prefix $10.1.i.0/24$. Consequently, router Y's routing table contains at least 255 entries corresponding to the network addresses 10.1.1.0/24, 10.1.2.0/24, ..., 10.1.255.0/24. Rather than advertising these 255 prefixes to router Z, route aggregation allows router Y to combine all of them into a single destination prefix 10.1.0.0/16, and announce only one route to Z.

When a router is configured to advertise an aggregate route, e.g., to destination prefix 10.1.0.0/16, it generates and advertises the aggregate route upon knowing at least one route to a more specific prefix, e.g., to 10.1.1.0/24. The more specific prefix is referred to as a child prefix, and the corresponding route a *child* or *contributing* route. Often in a network, some child prefixes of an aggregate prefix are not allocated to any subnets. Such child prefixes are commonly called *unallocated* or *unused* child prefixes.

While the primary application of route aggregation is to increase the Internet scalability by reducing routing table sizes, route aggregation is also used by operators to fulfill other requirements. For example, by restricting the scope of route advertisements, instabilities at the edge of a network

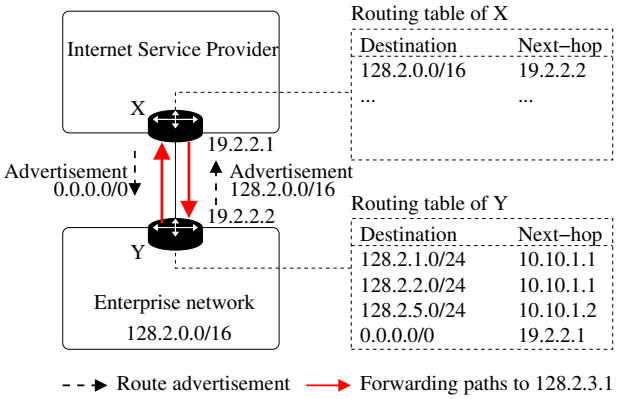


Figure 2: A persistent forwarding loop because of default route. Packets sent to the unallocated prefix 128.2.3.0/24 keep bouncing between X and Y.

are not propagated into the routing core. To illustrate, in Figure 1, let us assume that the interface connected to router X_1 and corresponding to 10.1.1.0/24 flaps. This hardware failure causes the router to continuously announce that interface alternately to be "up" and "down". With route aggregation deployed at router Y, such routing instabilities are hidden from router Z.

It is well known in the operator community that misconfiguration and mis-use of route aggregation may result in route anomalies such as persistent forwarding loops and blackholes.

Persistent forwarding loop for unallocated child prefixes [28]. Consider the scenario depicted in Figure 2. The enterprise network is allocated the address space 128.2.0.0/16. Router Y advertises an aggregate route for that address range to its ISP, which advertises the default route (i.e., 0.0.0.0/0) – an aggregate route also – back to the enterprise network. As an alternative, the enterprise network may be configured with a static default route to its ISP [8]. Suppose only some of the child prefixes of 128.2.0.0/16 (e.g., 128.2.1.0/24, 128.2.2.0/24, etc.) are allocated in the network. Now let's focus on an unallocated child prefix, 128.2.3.0/24. Packets sent to this prefix and arriving at router X are forwarded to router Y because of the aggregate route (128.2.0.0/16) advertised by Y. The traffic then reaches router Y which has no specific route to the destination. Hence, Y uses the default route received from X and forwards the traffic back to X, resulting in a persistent forwarding loop between routers X and Y.

Although unallocated prefixes may not directly impact user traffic, researchers have warned about attackers potentially exploiting this type of forwarding loops to cause network congestion [29].

To prevent this anomaly, operators commonly install a *sink route* in the router. It is a route corresponding to the aggregate address and pointing to the Null interface. Its goal is to drop all packets that match the aggregate address but

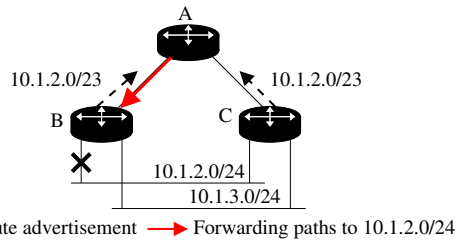


Figure 3: Illustration of a blackhole.

not a more specific route. For example, installing a static route for 128.2.0.0/16 pointing to the Null interface at router Y will prevent the above forwarding loop. Packets sent to 128.2.3.0/24 and reaching Y will not be sent back to X but dropped at Y. A prior study [29] has hypothesized the lack of sink routes (e.g., because of accidental omissions) to be at the origin of observed loops in the Internet. In response, router vendors have incorporated the feature of sink route creation into their latest router software. However, as we will show in Section 5, the current sink route solutions are not sufficient.

Blackhole [28]. Consider the scenario of Figure 3. Routers B and C are directly connected to two subnets: 10.1.2.0/24 and 10.1.3.0/24. Both routers are configured to advertise an aggregate route for 10.1.2.0/23 to router A.

Suppose the interface of B that connects to subnet 10.1.2.0/24 has just failed. Router B still announces the summary route 10.1.2.0/23 to router A because it knows a child route 10.1.3.0/24. As a result, router A receives two routes to 10.1.2.0/23, from B and C, respectively. Since the two routes have identical metrics, A may select B as its next-hop for packets sent to 10.1.2.0/24. However, upon arriving at B, these packets are dropped as B does not have a route to 10.1.2.0/24.

To conclude, this scenario shows that route aggregation can result in packets being blackholed despite the existence of a valid path to the destinations (e.g., A-C). Installing a sink route will not change the behavior.

3. CURRENT IMPLEMENTATIONS

Unlike most routing mechanisms (e.g., routing protocols), there is no single standard or IETF Request For Comment that precisely specifies the behaviors of route aggregation. Instead, route aggregation was introduced in conjunction with the Classless Inter-Domain Routing (CIDR) hierarchical addressing scheme, as a solution to (1) the exhaustion of IPv4 class B addresses, and (2) the explosion of routing table sizes. RFC 1338 – the IETF RFC that instituted route aggregation – only describes the main concept and general rules. For instance, RFC 1338 explains the key objective of route aggregation which essentially consists in aggregating the route information and advertising a single route in lieu of separately announcing all the child prefixes. RFC 1338 also defines several new rules: e.g., (1) forwarding must be performed based on the longest prefix match algorithm, (2) pack-

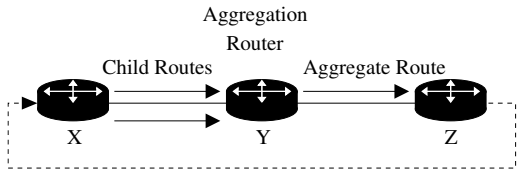


Figure 4: Setup for Experimenting with RA

ets that match an aggregate route but not any more specific routes must be discarded.

However, many questions related to RA are left unanswered, such as: When should routers aggregate route information? How is the aggregate prefix determined? On which interfaces should an aggregate be advertised? How is the metric of an aggregate route decided?

In practice, the exact RA behaviors are determined by the specific implementations of different routing protocols by different vendors [3, 18, 9, 11]. To understand the commonalities and differences among these implementations, we conduct a set of experiments over BGP, RIP, OSPF, and EIGRP that are implemented by Cisco and Juniper. We use Cisco 2600 (running IOS version 12.3) and Juniper 4300 (running JUNOS version 8.2R1.7) for our experiments.

3.1 Experiment Setup

Figure 4 depicts the setup for our experiments. Each experiment usually involves three routers. The first router, X, serves to generate child routes. We vary the number of child routes, the metrics of the child routes, their prefix lengths, and the protocol where they are advertised. The second router, Y, is the aggregating router. We configure it to perform route aggregation. We inspect Y's router FIB for the existence of sink route, and vary the administrative distance (AD) [10] of other running processes to determine the AD value of the sink route. When multiple routing protocols advertise routes to the same prefix, the router selects the route with the lowest AD value to install in its FIB [21]. Finally, the third router, Z, allows us to verify the advertisement and metric of the aggregate route. In a routing domain, a router may receive a route it initially advertised. In other words, the output of a router may impact its inputs. As such, we re-inject the aggregate route back to X.

3.2 Findings

Table 1 presents the summary of our findings. At the high level, the RA exhibits diverse behaviors on three aspects: modes of route aggregation, metric of aggregate route, and sink routes.

Modes of Route Aggregation: We find that there actually exists two main modes of route aggregation: *Auto-Summary* and *manual* aggregation, and that the latter can be further divided into several sub-classes: interface, area/level, BGP AS, instance, and router-based manual aggregation. For all of them, the presence of a child route is a necessary and sufficient condition to initiate the advertisement of an aggregate route. However, each type presents distinct characteristics:[1]

```
1   routing-options {
2       aggregate {
3           10.1.0.0/23;
4       }
5   }
6   policy-options {
7       policy-statement aggregate-into-rip {
8           term first-term {
9               from protocol aggregate;
10              then accept;
11          }
12      }
13  }
14  protocols {
15      rip {
16          export aggregate-into-rip;
17      }
18  }
```

Figure 5: RA configuration for Juniper.

Interface. This mode of route aggregation allows routing processes to advertise an explicitly configured aggregate route, instead of the child routes, out of specific interfaces.

Area/Level. Link-state routing protocols flood link state information (not routes) to all participating routing processes. However, OSPF and IS-IS also offer the notion of areas, and levels respectively, to allow hierarchical designs, and between areas/levels, routes are exchanged in a vectoring manner and hence can be aggregated.

Router/Instance. For Juniper routers, route aggregation is configured in a consistent way for all routing protocols. The configuration consists of three steps (Figure 5): The first one (lines 1 to 5), begins with a *routing-options* statement, and creates a sink route. Then, a *policy-options* statement (lines 6 to 13) defines the export policies for the sink route. Finally, the export rules are applied to the protocols (lines 15 to 18) where the sink route is to be advertised. A child route must be present in the FIB for a sink route to become valid and to be considered by the route selection procedure. The creation of the sink route is therefore per router, while the advertisement is per routing instance.

Metric of Aggregate Route: The metric of the aggregate route is determined in a variety of ways. It is set to the minimum or maximum of all the child prefixes depending on the routing protocol and vendor. We note that for many implementations (e.g., Cisco RIP, Cisco EIGRP), the method the metric of the aggregate route is computed cannot be modified. For BGP, route advertisements include an AS-PATH attribute, which is an ordered sequence of the autonomous systems the route has traversed. For Cisco routers, the AS-PATH of aggregate routes is by default reset to the aggregating network. However, optionally, a router can also include an AS-SET attribute which indicates the autonomous systems where the child prefixes came from.

[1]The *auto-summary* and *BGP-AS* modes are described in [23].

Vendor	Routing Protocol	Modes of Route Aggregation		Metric of Aggregate Route	Sink Route	
		Auto-Summary	Manual		Auto-Creation	Default AD
Cisco	EIGRP	Yes	Per Interface	min()	Yes	5
	RIP	Yes	Per Interface	min()	No	N/A
	OSPF	No	Per Area	min(), later max()	Yes	110
	BGP	Yes	Per AS	Origin/AS-SET	Yes	200
Juniper	*	No	Creation Per Router Advertisement Per Instance	Customizable	Yes	130

Table 1: Characterization of Route Aggregation behaviors.

Sink Route: Most implementations automatically create a sink route upon advertising an aggregate route. However, the default AD value of an aggregate route may be higher than that of routing protocols. For example, for Juniper, the administrative distance value of aggregate routes is by default 130. In contrast, the default administrative distance values of internal OSPF routes, IS-IS Level 1 internal routes, IS-IS Level 2 internal routes and RIP are respectively 10, 15, 18 and 100. As such, the sink routes may not get installed in a router's FIB. Finally, we note that certain implementation (e.g., Cisco RIP) neither creates any sink route nor offers a mean for operators to install one.

4. A MODEL FOR ROUTE AGGREGATION

The previous section revealed the ad-hoc nature of the current design and implementations of route aggregation (RA). The results motivated us to develop an analytical model to reason about RA, starting from the hypothesis that the functionality can be succinctly defined by a small number of simple functions and their interactions with other routing components of a router. Furthermore, prior work has attributed RA to routing anomalies such as forwarding loops [29] and blackholes [28]. A unified model of route aggregation, by abstracting away many of the implementation details, would also help identify the real root causes of these anomalies.

In this section, we present such a unified analytical model. First, we introduce two router level RA specific primitives and incorporate them into a canonical router model. The observed diverse RA behaviors can be captured entirely by the action of these simple primitives and their interaction with other routing components on the same router. Then, we illustrate how the model may be used to predict the router FIB content and the route advertisements at each router.

4.1 Route Aggregation Primitives

Based on the experimental results, we observe that the essence of route aggregation lies in two software primitives which we term *add-sink()* and *adv-aggr()*. Cisco implementations rely on both while Juniper implementations use only the first primitive.

4.1.1 add-sink()

The *add-sink()* primitive takes two input arguments and outputs a set of sink routes to the route selection procedure.

Primitive 1 *add-sink(E, A)*

Input: (1) E - routes from the router FIB or routes from a protocol specific Route Information Base (RIB); (2) A - all aggregate routes configured on the router

1: $S = \{\}$
2: Remove existing sink routes from E
3: **for all** $a \in A$ **do**
4: **if** there exists a child route of a in E **then**
5: Set AD value and Null next hop for a;
6: Add a to S;
7: **end if**
8: **end for**
9: Present S to the route selection procedure

The first argument E is a set of routes present at the router. However, depending on the implementations, this set may correspond to the routes in either the FIB (for JUNOS), or a protocol specific routing information base (RIB) (for Cisco). The second argument A is the set of all aggregate routes that are configured on the router.

The *add-sink()* primitive embodies two key characteristics. First, while all vendor implementations except Cisco RIP create a sink route upon knowing a child route of a configured aggregate route, not every sink route created is installed in the FIB. Each sink route, upon creation, is assigned its own administrative distance (AD) value (Table 1) – e.g., sink routes created from an EIGRP routing process is set a default AD value of 5. Inside the route selection procedure, the sink route will compete with other routes to the same prefix that may be offered by different routing processes (e.g., OSPF, BGP, or static), and unless the sink route has the lowest AD value in the group, it will not be installed in the FIB.

Second, the location where the primitive examines for the presence of child routes differs depending on the implementation. As explained in Section 3, for Juniper routers, the creation of sink route is per router and the presence of a child route in the FIB is a necessary and sufficient condition for a sink route to be created. In contrast, we discovered that the presence of child routes in the FIB is not a sufficient condition for Cisco routers [23]. The sink route is created and present only when a child route is present in the $RIBout$.

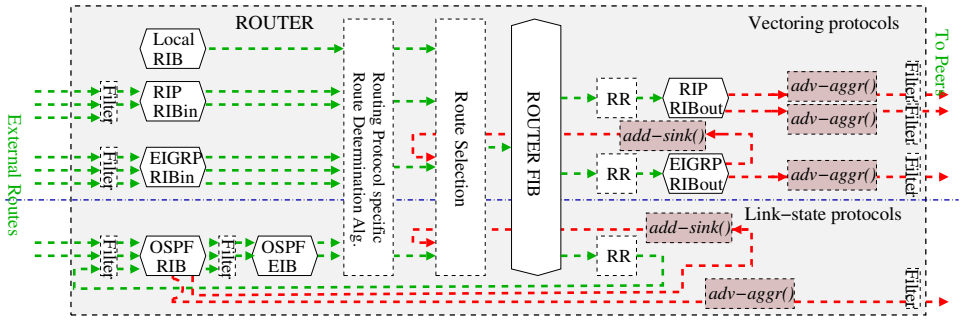


Figure 6: Per-router model of RA in Cisco. The essence of RA lies in two main primitives *add-sink()* and *adv-aggr()*. Both primitives are applied to the RIBout/RIB of the routing processes. *add-sink()* is not implemented for RIP. The router model for Juniper implementations is provided in [23]: Specifically, *add-sink*() takes input from only the router FIB, and *adv-aggr*() is not implemented. Instead, the router relies on route redistribution to advertise the aggregate routes.

Primitive 2 *adv-aggr(E, A)*

Input: (1) E - routes from a protocol specific RIBout; (2) A - all aggregate routes configured on given interface

1: Remove sink routes from E
2: **for all** $a \in A$ **do**
3: **if** there exists a child route of a in E **then**
4: Remove all child routes of a in E;
5: $a.m = metric(a, E)$;
6: Add a to E;
7: **end if**
8: **end for**
9: Advertise E on the interface

4.1.2 *adv-aggr()*

The second primitive, *adv-aggr()* handles the advertisement of aggregate routes to the router's peers. JUNOS implementations rely on *export* policies to announce the aggregate routes from the FIB into the routing processes. (See Section 3.) As such, JUNOS routers simply rely on route redistribution [21] to advertise the aggregate routes. In contrast, Cisco implementations depend on a separate primitive that we call *adv-aggr()*.

As highlighted in Section 3, operators may configure different aggregate routes on different interfaces. Consequently, *adv-aggr()* is performed for each interface and per routing process. The primitive takes two input arguments: E, the set of routes present in a *RIBout* – the part of a RIB for storing routes to be advertised out, and A, the set of aggregate routes configured on a given interface. It determines and advertises a set of aggregate routes on that interface.

To determine the set of routes to advertise, *adv-aggr()* first removes all sink routes from E. It then goes through all the aggregate routes configured on the interface, determines a subset with a least one child route included in E, and advertises this subset of aggregate routes on the interface. The metric of the advertised aggregate routes is set by the *metric()* function, which is routing process specific. (See Table 1.) For example, metric(a, E) returns, for OSPF, the maximal metric of all a's child routes in E.

4.2 Canonical Router Model

A prior study [22] has proposed a canonical router model to study route selection and route redistribution (RR), the procedures that govern how a router ranks and exchanges routes received from different routing protocol processes. We have successfully incorporated the new route aggregation primitives into the router model. Figure 6 depicts the extended model for Cisco inplementations.

The new router model allows us to reason about how the route aggregation primitives interact with a router's other routing components to impact the content of the router FIB, and thus influence, directly, how packets are routed at the router. In particular, we note that the large diversity of observed RA behaviors can be explained by the differences in where the primitives *add-sink*() and *adv-aggr*() are applied in different implementations. For brevity, we defer the details to a companion technical report [23].

4.3 Router FIB Prediction

The canonical router model allows us to infer whether a route is in the FIB and how it may be advertised at a router, by tracing the steps of the route moving through the router. Using this methodology, we have discovered a surprising and counterintuitive result as follows.

Unexpected route loss: *Configuration of route aggregation on an interface can prevent the advertisements of routes on other interfaces.*

To illustrate it, we hypothesize a Cisco router running EIGRP on four interfaces I_1, I_2, I_3 and I_4. We assume that the router receives only two routes: r_1 and r_2 to destination prefixes 192.168.1.0/24 and 192.168.0.0/16, respectively on interface I_1. Furthermore, we assume that an aggregate route r_3 to 192.168.0.0/16 is configured on I_2 and another aggregate route r_4 to 192.0.0.0/8 is configured on I_3. As illustrated in Figure 7, the interaction of these routes would produce the following sequence of events:

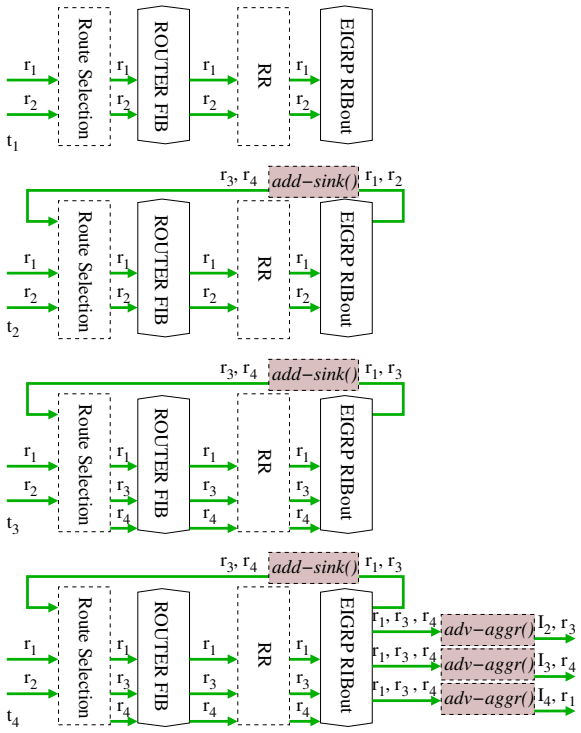


Figure 7: Configuration of aggregate routes on interfaces I_2 and I_3 impacts the announcements on interface I_4. Although the router receives two routes r_1, r_2 to different prefixes, only one of the routes (r_1) is advertised on I_4.

t_1 The two received routes, r_1 and r_2, are installed in the FIB and the EIGRP *RIBout*.

t_2 Then, after executing *add-sink(*$\{r_1, r_2\}, \{r_3, r_4\}$*)*, the output $\{r_3, r_4\}$ is presented to the route selection procedure.

t_3 Because r_3 has an AD value of 5 whereas r_2 has an AD value of 90, r_3 is preferred and installed in the FIB, serving as a sink route. As for r_4, it is the only route to 192.0.0.0/8 and is therefore also installed in the FIB.

The EIGRP *RIBout* is updated. r_2 is removed, while r_3 and r_4 are added. *add-sink(*$\{r_1, r_3, r_4\}, \{r_3, r_4\}$*)* is executed, passing $\{r_3, r_4\}$ to the route selection procedure.

t_4 *adv-aggr()* is performed for each interface.
For I_2, *adv-aggr(*$\{r_1, r_3, r_4\}, \{r_3\}$*)* returns r_3. As such, r_3 is advertised out on I_2.
For I_3, *adv-aggr(*$\{r_1, r_3, r_4\}, \{r_4\}$*)* returns r_4. Finally, on I_4, *adv-aggr(*$\{r_1, r_3, r_4\}, \{\}$*)* returns r_1.

While the advertisements on interfaces I_2 and I_3 are as expected, the announcements on I_4 is surprising. No route aggregation is configured on I_4. When the router receives two routes r_1 and r_2 to different prefixes, one may expect both routes to be advertised out of I_4. However, it turns out that only r_1 is announced. r_2 has been "filtered out" and is lost.

We implemented the above scenario with actual Cisco routers and confirmed the route loss.

5. ANOMALIES AND ROOT CAUSES

Using the new model of route aggregation, we have performed an in-depth analysis of routing anomalies that may result from route aggregation. We already described one of the results, i.e., unexpected route loss, in the previous section. In this section, we present the other key results from the analysis. First, we identify the root causes for the known anomalies, namely forwarding loops and blackholes. Then, we disclose three new additional routing anomalies that may occur with route aggregation and analyze their root causes. Table 2 contains a quick summary of our results on routing anomalies.

Our main finding is that the current design of sink route creation does not guarantee the installation of the sink route in the FIB, i.e., the sink route is created but has no effect. This may give a false sense of safety. In addition, sink routes do not mean anomaly-free routing: we showed that sink route can also cause route loss.

Our methodology has two major elements. The first is a step by step simulation of the creation and movement of aggregate and sink routes, with the aid of the canonical router model. The other is leveraging the recently proposed Metarouting theory [16]. As described in Section 3, route aggregation is relevant and can happen only when routes are iteratively advertised router by router, as with vector routing protocols. For such protocols, the Metarouting theory establishes that strict monotonicity (SM) is a sufficient condition for convergence to loop-free paths [16]. Put it simply, the SM property stipulates that a route's preference should strictly decrease when propagated in the network. Therefore, we specifically looked for scenarios that may violate the SM condition when looking for new anomalies.

Our goal is not to enumerate all the possible routing anomalies that could be caused by route aggregation. Neither do we have evidence that these anomalies are happening in the real world. Despite these, we believe our findings would provide important insights for studying routing anomalies, developing better practice and guidelines to ensure routing safety, and designing next generation routing mechanisms that are more robust and safe.

5.1 Root Causes of Known Anomalies

Forwarding loops for unallocated child prefixes. Recall that a scenario exhibiting this anomaly was illustrated in Figure 2. It involves a route aggregation implementation (Cisco RIP) that does not implement the *add-sink()* primitive. Without a sink route, a router may advertise an aggregate route with "holes", i.e., the router FIB has no entries for all the unallocated child prefixes. Consequently, when receiving packets destined to those prefixes, the router may return them back to the previous hop (e.g., because of the default route in the example scenario), causing a persistent forwarding loop.

We note that advertising an aggregate route without a sink route effectively propagates nonexistent routes for the un-

Anomaly	Status	Root Cause
Forwarding Loop (unallocated prefixes)	Previously known	No sink route is created; or sink route fails to install in the FIB.
Blackhole	Previously known	Advertisement of a child prefix without a route other than the sink route.
Forwarding Loop (allocated prefixes)	Newly discovered	No sink route is created; or sink route fails to install in the FIB.
Route Oscillation with Count-to-Infinity	Newly discovered	Interaction between multiple aggregating routers that fail to install a sink route in the FIB.
Count-to-Infinity (perpetual)	Newly discovered	No sink route is created; or sink route fails to install in the FIB.
Route Loss	Newly discovered	Sink route overriding another valid route for the same aggregate prefix.

Table 2: Summary of Possible Routing Anomalies from Current Design of Route Aggregation.

used prefixes. Nonexistent routes should be the least preferred in the network. Propagating them is certainly a violation of the SM condition. A corollary of this conclusion is that this category of anomalies may be wider because even with *add-sink()* implemented, not every sink route generated is installed in the FIB; as described in the previous section, each sink route needs to compete with routes from other routing processes based on the AD value. We substantiate this corollary later in the section.

Blackholes for some child prefixes. Recall that a scenario for this anomaly was illustrated in Figure 3. This problem occurs partly because of the introduction of sink routes. When a router advertises an aggregate route with "holes", packets destined for one of the locally unknown child prefixes may arrive unexpectedly and as a result, get discarded due to the sink route. If that child prefix is unallocated throughout the entire network, the router in fact does the right thing: Packets to unallocated prefixes should be dropped as soon as possible. However, if the child prefix is allocated but just unknown to this particular router (e.g., because of failures), we have a blackhole. Therefore, the occurrence of blackhole is also largely due to an event happening on another router: the advertised aggregate route is mistakenly preferred over all routes that point to a legitimate forwarding path to that allocated child prefix. Since the root cause involves multiple routes, the SM condition does not apply here. In this way, blackholes may be more difficult to detect and prevent than other anomalies.

5.2 New Anomalies

Our analysis has uncovered a total of four new types of routing anomalies. We validated all the presented anomalies in our test bed, with the exception of the count-to-infinity of arbitrary time length as this anomaly requires specific race conditions that are difficult to produce in our environment.

One of the the anomalies was already presented in Section 4. The remaining three anomalies fall into two groups based on their root cause. The first group of anomalies is caused by the absence of a sink route in the FIB of a single router. The other class results from the interaction between multiple routers that advertise the same aggregate route. We emphasize that the example scenarios below involve both

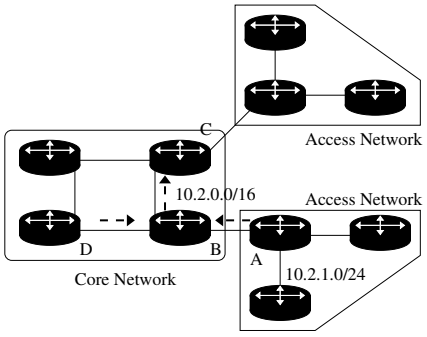


Figure 8: Illustration of perpetual count-to-infinity.

Cisco and Juniper routers, with default settings for routing protocols. Therefore, we believe there is a good chance they will happen in practice.

5.2.1 Absence of Sink Route in One Router FIB

As discussed in Section 4, there are two cases where a sink route is not installed in the router FIB. First, some vendor implementations do not implement the *add-sink()* primitive, and thus do not install a sink route. Second, a sink route may be assigned an AD value that is larger than that of another route learned by a routing process, and thus is not chosen by the route selection procedure. Below, we show a new routing anomaly that is a consequence of the latter case.

Perpetual Count-to-Infinity. Consider the network depicted in Figure 8. Enterprise networks are usually designed according to a hierarchical structure with a core connecting all access networks. Access networks commonly have a hub and spoke topology, whereas the core may consist of a ring to a full mesh depending on the financial constrainsts and resiliency requirements. We assume router B to run JUNOS and to be configured to advertise via RIP an aggregate route for 10.2.0.0/16 out of its interface to router C. The following events lead to a perpetual count to infinity problem.

t_1 Router A has a directly connected subnet 10.2.1.0/24, and advertises this prefix to B.

t_2 Router B receives the announcement from A and installs a route to 10.2.1.0/24 in its router FIB. Then, as this route is a child route of 10.2.0.0/16, B advertises an aggregate

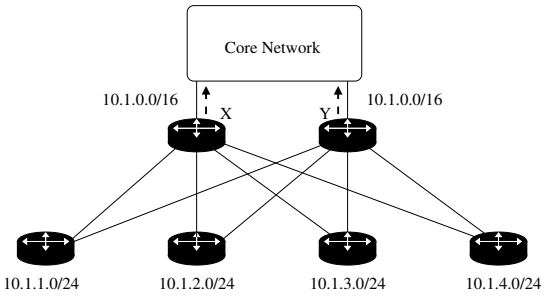


Figure 9: Route oscillations.

route for 10.2.0.0/16 to router C, with a metric value of 1.

t_3 The route gets propagated and comes back to router B, from router D, with a metric value of 4. In scenarios where this route has a lower AD value than the sink route, which is the case with JUNOS default setups (100 for RIP routes vs. 130 for a sink route), the route selection procedure prefers it and installs it in the FIB in place of the original sink route. Since the route to 10.2.0.0/16 (at router B) now has a metric value of 4, B sends a new advertisement to C with a metric value of 5.

The advertisement cycle continues and the metric of this route keeps incrementing until it reaches the maximum allowed value. The route is then discarded, a sink route is installed in the FIB again, and the whole process repeats. We have a perpetual count-to-infinity problem. This anomaly is particularly undesirable because the network is permanently unstable.

5.2.2 Interaction of Multiple Aggregating Routers

Route oscillations may derive from the interaction of multiple aggregating routes which creates a race condition similar to that of a BGP dispute wheel [15]. In addition, our analysis reveals that route aggregation can cause persistent forwarding loops for allocated child prefixes, not just for unallocated child prefixes as described in Section 2. Because of space limitations, we only present the route oscillations. For the forwarding loops for allocated addresses, we refer the reader to [23].

Route oscillation with count-to-infinity. Consider the network depicted in Figure 9. For added resiliency, access networks are frequently connected to the core through multiple border routers. Routers X and Y are Juniper routers. The configuration is such that routers X and Y both advertise an aggregate route for 10.1.0.0/16 with the same metric values. The following sequence of events leads to route oscillations.

t_1 Routers X and Y both advertise the aggregate route and at the same time, create a sink route. Because the sink route is the only available route to 10.1.0.0/16, the sink route is installed in each router's FIB.

t_2 Router X receives the aggregate route advertised by router Y. X now has two routes to 10.1.0.0/16: the local sink route, and the route received from Y.

For scenarios where the route received from Y is more preferred to the local sink route (because of their AD value assignments), the sink route is removed from the FIB. As a result, router X stops advertising the aggregate route and instead, re-advertises the route from Y after incrementing its metric. Router Y may concurrently perform the same actions, preferring the route from X to the destination 10.1.0.0/16, and re-advertising it with a larger metric value.

t_3 If X and Y are synchronized in processing their routing information, the following cycle repeats at each router: receiving a route with a larger metric value than the route in its FIB, updating its FIB with the new route, and re-advertising the new route after incrementing the metric. It ends when the maximum metric value is reached. The two routes are then withdrawn, and the network is back to the same state as at t_1.

We obtain a route oscillation in which routers X and Y periodically switch back to the sink route for a short period and for the rest of the time, keep selecting and advertising each other's route in a count-to-infinity fashion. The anomaly may last for an arbitrary amount of time, until the synchronization between X and Y stops.

Interestingly, if Cisco routers were used for X and Y, there would be a persistent forwarding loop for unallocated child prefixes. This is because the Cisco specific *adv-aggr()* primitive would advertise an aggregate route with the same metric value (i.e., minimal metric of all child routes) as long as the child route is in the RIBout. So at step t_2, both X and Y would still advertise an aggregate route with the same metric value as at t_1. There is no count-to-infinity problem. However, because the sink routes are squeezed out of the router FIBs, we have a persistent forwarding loops for unallocated child prefixes.

6. ANALYSIS OF EXISTING SOLUTIONS

Router vendors have proposed several guidelines to deal with the known anomalies caused by route aggregation. In this section, we explain why they fall short in solving the problem. Furthermore, we show that it is not always possible to detect anomalies early, by statically checking route configurations alone. These negative results indicate the limitation of patchy solutions, and they have motivated us to formulate conditions on the intrinsic behaviors of the two route aggregation primitives that can guarantee anomaly-free routing regardless of configuration. (We will present the conditions in Section 7.)

6.1 Inadequacy of Vendor Guidelines

Installation of sink routes. The most widely-known vendor guideline is the one stipulated by RFC 4632 [13], which

defines the following rule for routers performing route aggregation:

A router that generates an aggregate route for multiple, more-specific routes must discard packets that match the aggregate route, but not any of the more-specific routes. In other words, the "next hop" for the aggregate route should be the null destination. This is necessary to prevent forwarding loops when some addresses covered by the aggregate are not reachable.

This rule makes sense as a correctness criterion for route aggregation behaviors. How to meet the criterion is far from straightforward. For example, an early version of the RFC [14] suggested to create sink routes. However, the creation of a sink route alone does not guarantee the route to be active, i.e., included in the router FIB, as illustrated in the scenarios presented in Section 5. Moreover, the installation of a sink route at a router may create a blackhole for some child prefixes that are valid even though they are not explicitly known at that particular router.

Topology specific guidelines. Cisco documentation [2] describes multiple scenarios where blackholes may result from route aggregation, and suggests two main approaches to address the issue. Both approaches require a specific topology to work. The first is based on adding a physical link between two aggregating routers that are gateways of a common child prefix (e.g., router B and C in Figure 3.) Doing so would allow one gateway, after losing its direct connection to the child prefix, to learn a route to that prefix from the other gateway, thus preventing the blackhole. The second approach places an even stronger restriction on the network topology, only permitting route aggregation by a router that is the only exit point (i.e., a choke point) between two segments of the network.

6.2 Difficulty of Early Detection of Anomalies

Given the limitation of the vendor guidelines, a natural question arises: Is it possible to detect anomalies early using static analysis techniques that have proven effective in several areas of software development. Unfortunately, the answer to this question is not a positive one.

Theorem 1: *The problem of determining whether the collection of route aggregation configurations in a network is vulnerable to persistent forwarding loops is NP-complete.*

The proof mainly relies on a polynomial reduction of the the well known NP-complete problem of 3-CNF SAT. The details are omitted because of space limitations, but can be found in [23].

7. NEW SAFETY CONDITIONS

So far, we have exposed a slew of problems with the current route aggregation design and the difficulties of addressing these problems without changing the behavior of the primitives. In the following two sections, we look beyond the current design and explore alternative approaches. Ideally, a new design should satisfy all these four safety properties: (1) Convergence (i.e., no route oscillations), (2) Loop-free packet forwarding paths, (3) No blackholes, and (4) No unexpected route losses.

In the first alternative approach, we focus on modifying just the behavior of the add-sink() and adv-aggr() primitives. Specifically, we formulate a set of safety conditions on their new behavior.

It has proven a challenge to formulate one set of conditions for the primitives to guarantee all four properties. The challenge lies in the fact that the prevention of some of the anomalies like blackholes may require either (i) network-wide coordination among the routers in addition to a new design of the primitives, or (ii) a very stringent requirement on when a router can advertise an aggregate route. Also, the conditions should be general, independent of routing protocols, router vendor architectures, and the network topology. As a result, the conditions presented below may be too stringent to meet some of the existing design goals.

7.1 Convergence to Loop-Free Paths

Condition 1. *For an aggregate prefix, a sink route is added and always possesses the lowest unique AD value.*

Condition 1 is sufficient to guarantee convergence and loop-free forwarding paths because it preserves the strictly monotonic (SM) property [16]. When a sink route is present and always has the lowest AD value, it is preferred to the advertised aggregate route, ensuring the preservation of SM.

Due to space limitations, a detailed proof of the correctness of this condition, which can be found in [23], is omitted.

7.2 Blackhole Prevention

Condition 2. *A router advertises an aggregate route only if a set of routes that fully covers the address space of the aggregate route is present in the FIB.*

Condition 2 guarantees not only convergence and loop-free forwarding paths, but also the absence of blackholes. The first part can be proven through the preservation of the SM propery, while the blackhole prevention part can be demonstrated by contradiction. The details can be found in [23].

This condition directly addresses the problem illustrated in Figure 3: Compliance with Condition 2 ensures that router B will stop advertising the aggregate route and announce a route to 10.1.3.0/24 instead when it no longer has a route to subnet 10.1.2.0/24. Consequently, router A will forward all packets destined to that subnet only to router C, and there will be no chance for the packets to be blackholed at router B.

We note that router C can still advertise the aggregate route. In addition, for a network with a hierarchical routing

design structure as recommended by vendors [26], routers in a higher tier (e.g., router A in Figure 3) may still be able to advertise aggregate routes. Clearly, further empirical studies of both intra-domain and inter-domain topologies are needed to validate the hierarchical structure of the Internet, and assess whether this condition may be too stringent in practice. Section 8 describes an alternative design for route aggregation that eliminates blackholes without requiring Condition 2.

7.3 Prevention of Route Loss

Condition 3. *add-sink() and adv-aggr() should not create a new route in the presence of another route in the router FIB advertising the same prefix as the aggregate.*

Condition 3 eliminates the type of route losses identified in Section 4, and can be combined with either Condition 1 or Condition 2. If *add-sink()* and *adv-aggr()* create no new route, the route that was lost before now has no competition, and therefore continues to stay in the FIB and be advertised. (Again, the details of the proof can be found in [23].) Note that *adv-aggr()* should still suppress child routes of the aggregate prefix in this case.

8. A CLEAN SLATE APPROACH

In this section, we no longer restrict ourselves to the existing primitives but explore a bolder design alternative for route aggregation. First, we revisit the role and the fundamental semantics of route aggregation.

Route aggregation reduces the storage requirements for the forwarding tables by leveraging the hierarchical structure of IP addressing and allowing routers to treat multiple destination networks collectively as a single logical destination.

However, while a route implies all destinations included in the advertised prefix to be reachable, the current design of route aggregation violates this assumption. As shown in Sections 2 and 5, an aggregate route may include sub-blocks of addresses that are not reachable. This inconsistency between views of the control and data planes may be the ultimate culprit of the routing anomalies and blackholes.

Therefore, to mitigate all the problems, we introduce the novel concept of *negative route*. In particular, routers must advertise and store in their forwarding table, in conjunction of an aggregate route, child prefixes for which they do not have a route.

In relation to the results of the previous section, the use of negative routes would make Condition 2 unnecessary. A router no longer requires all child prefixes to be present in order to advertise an aggregate route. Instead, routers will use negative routes to precisely define the prefixes for which they have a valid route. Consequently, other routers can then select only valid routes to target destinations.

To illustrate it, we assume a network making use of the 192.168.0.0/16 block of IP addresses: A router that can reach all the /24 child prefixes has advertised a single prefix 192.168.0.0/16. Suppose that due to a link failure, one child prefix 192.168.255.0/24 becomes unreachable. Condition 2 requires a new set of aggregate routes to be computed as follows. All remaining reachable /24 child prefixes will be merged into maximal sized blocks with a common prefix. As such, in the absence of negative routes, in order to comply with Condition 2, the router has to advertise 8 prefixes: 192.168.0/17, 192.168.128/18, 192.168.192/19, 192.168.224 /20, 192.168.240/21, 192.168.248/22, 192.168.252/23, 192.168.254/24. In contrast, with the capability of negative routes, that router would only handle two routes: 192.168.0/16 (aggregate) and 192.168.255/24 (negative).

Two prior proposals have semblance of negative routes. First, routers currently advertise invalid routes (e.g., RIP route with a distance of 16) but the goal of such routes is only to withdraw previously advertised routes. Upon receiving an invalid route, a router removes the route if it is present in the routing table. In contrast, our notion of negative route has stronger semantics than that of an invalid route. A negative route should be stored in a router's routing table, and taken into consideration when selecting a best route for a destination. Doing so eliminates all routing anomalies due to unreachable sub-prefixes in an aggregate route.

The other related proposal is the IS-IS Detailed IP Reachability Extension [12], which suggests route advertisements to be accompanied by a bit vector. Each bit represents the reachability to one address in the range of addresses covered by the route. As such, a router could learn the exact "holes" in a summary address. However, the extension was designed to solve a different problem: that of the reachability of BGP next hops for the BGP next hop tracking feature. As such, [12] specifically stipulates that: "*the information present in the detailed reachability sub-TLV should not be used to generate any dataplane forwarding entry.*"

In other words, the proposal does not address the aforementioned inconsistency between the control and data planes. To illustrate this shortcoming, let us revisit the blackhole problem depicted in Figure 3. Suppose that router B has lost its connection to 10.1.2/24, and indicated this change to router A through the bit vector method. As recommended by router vendors [26], router A, being in a higher tier, may advertise only the default route to routers B and C. In that case, C will not be able to react to what has happened at B, e.g., by advertising a more specific route to 10.1.2/24. Further, since A does not adjust its forwarding table based on the bit vector advertised by B, it will still send packets destined to 10.1.2/24 to B, resulting in a blackhole. In contrast, by allowing routers to advertise and store negative routes, our proposal ensures a consistent and accurate view of reachable destinations at each router.

Finally, this approach may also enable new TE techniques for multi-homed networks. We defer this topic, as well as an analysis of the required modifications to the routing table structures and the route selection logic, to future work.

9. RELATED WORK

The IRTF Routing Research Group [5] has been examining a number of long term proposals to improve the scalability of the Internet routing system. The proposals can be classified into two main approaches: separation or elimination. Both approaches require significant modifications to the existing infrastructure. Separation based schemes (e.g., [27]) rely on a new mapping infrastructure. Elimination based proposals (e.g., [7]) require major changes to end systems. In addition, routing paradigms such as compact routing [20] have been introduced to minimize the memory requirement for storing routing information. In comparison to these proposals, our main focus is to expose the risks of the current route aggregation design and explore less disruptive solutions that do not require major changes to existing routing protocols or end systems.

Several short term proposals [30, 6] focus on reducing the size requirement of a router FIB. While they may improve router performance, they have a limited impact on how routes are advertised by routing protocols and as such, they do not address anomalies that can result from route aggregation, which is often used for privacy reasons too.

10. CONCLUSION

We make several contributions in this paper. First, we perform a study and show that the RA behaviors vary significantly across routing protocols and router vendors even for simple network setups. We then present a model that captures the diverse RA behaviors as in today's router implementation and the interactions between router aggregation and other functional blocks in a router. The model enables us to perform a number of analysis for a single router and a network of routers. We show that existing RA configuration guidelines do not guarantee safety. For example, configuration of sink routes is a key recommendation for avoiding persistent routing loops with RA and a previous study assumes that the persistent routing loops observed was due to lack of configuration of sink route. However, we show that configuration of sink routes may not result in the installation of sink routes in the FIB and may also cause unexpected route loss. Our analysis also finds that RA is tightly intertwined with other essential functional components of a router (e.g., route selection, route redistribution), and these complex interactions can lead to unexpected outcomes. We further prove that determining whether a network configuration with route aggregation can result in a persistent forwarding loop is NP-complete. Finally, we identify a set of sufficient conditions to guarantee routing safety, and explore clean-slate designs for this fundamental primitive.

11. ACKNOWLEDGMENTS

We thank Robert Beverly, Olivier Bonaventure, Joao Sobrinho, and our shepherd Amund Kvalbein, for helpful discussions and suggestions. This research was partially sponsored by the NSF under the 100x100 Clean Slate Project (NSF-0331653), the 4D Project (NSF-0520187), grants CNS-0520210, CNS-0721574 and a Graduate Research Fellowship. Views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of NSF, or the U.S. government.

12. REFERENCES

[1] BGP Routing Table Analysis Reports. http://bgp.potaroo.net/.
[2] Cisco IOS IP Configuration Guide, Release 12.2.
[3] Cisco IOS Release 12.0 Network Protocols Configuration Guide.
[4] Internet Systems Consortium. www.isc.org.
[5] IRTF Routing Research Group. www.irtf.org.
[6] H. Ballani, P. Francis, T. Cao, and J. Wang. Making Routers Last Longer with ViAggre. In *NSDI*, 2009.
[7] J. Bi, P. Hu, and L. Xie. Shim6: Reference Implementation and Optimization. In *Networking, pages 302-313*, 2008.
[8] R. Bush, O. Maennel, M. Roughan, and S. Uhlig. Internet optometry: Assessing the broken glasses in internet reachability. In *IMC*, 2009.
[9] Cisco. Understanding Route Aggregation in BGP, August 2005.
[10] Cisco. What is Administrative Distance?, March 2006.
[11] Cisco. IP Summary Address for RIPv2, Retrieved January, 2010.
[12] C. Filsfils, S. Previdi, and G. Swallow. IS-IS Detailed IP Reachability Extension, 2008. Internet Draft draft-swallow-isis-detailed-reach-01.txt.
[13] V. Fuller and T. Li. Classless Inter-domain routing (CIDR): The Internet Address Assignment and Aggregation Plan, 2006. Request for Comments 4632.
[14] V. Fuller, T. Li, J. Yu, and K. Varadhan. Supernetting: an Address Assignment and Aggregation Strategy, 1992. Request for Comments 1338.
[15] T. Griffin, F. B. Shepherd, and G. T. Wilfong. The stable paths problem and interdomain routing. *IEEE/ACM Trans. Netw.*, 2002.
[16] T. G. Griffin and J. L. Sobrinho. Metarouting. In *Proc. ACM SIGCOMM*, 2005.
[17] G. Huston. Analyzing the Internet's BGP Routing Table, January 2001.
[18] Juniper. JUNOS Internet Software for J-series, M-series, and T-series Routing Platforms Routing Protocols Configuration Guide, Retrieved January, 2010.
[19] E. Katz-Bassett, H. V. Madhyastha, J. P. John, A. Krishnamurthy, D. Wetherall, and T. Anderson. Studying Black Holes in the Internet with Hubble. In *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2008.
[20] D. Krioukov, kc claffy, K. Fall, and A. Brady. On compact routing for the internet. In *ACM CCR*, 2007.
[21] F. Le, G. Xie, and H. Zhang. Understanding Route Redistribution. In *Proc. IEEE ICNP*, 2007.
[22] F. Le, G. Xie, and H. Zhang. Instability Free Routing: Beyond One Protocol Instance. In *Proc. ACM CoNEXT*, 2008.
[23] F. Le, G. G. Xie, and H. Zhang. Understanding Route Aggregation. In *Technical Report, Carnegie Mellon University, CMU-CS-10-106*, 2010.
[24] G. Malkin. RIP Version 2, 1998. Request for Comments 2453.
[25] D. Meyer, L. Zhang, and K. Fall. Report from the IAB Workshop on Routing and Addressing, 2007. Request for Comments 4984.
[26] P. Oppenheimer. *Top-Down Network Design (2nd Edition)*. Cisco Press, 2004.
[27] L. Wang, D. Jen, M. Meisel, B. Zhang, H. Yan, D. Massey, and L. Zhang. Towards A New Internet Routing Architecture: Arguments for Separating Edges from Transit Core. In *ACM Workshop on Hot Topics in Networks (HotNets-VII)*, 2008.
[28] R. White, D. Slice, and A. Retana. *Optimal Routing Design*. Cisco Press, 2005.
[29] J. Xia, L. Gao, and T. Fei. A Measurement Study of Persistent Forwarding Loops on the Internet. In *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 2007.
[30] X. Zhao, Y. Liu, L. Wang, and B. Zhang. On the Aggregatability of Router Forwarding Tables. In *INFOCOM*, 2010.