BGP ROUTE FLAP DAMPING ALGORITHMS

by

Wei Shen
B.Sc., University of British Columbia, 2002
B.A.Sc., East China University of Science and Technology, 1989

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

In the School
of
Computing Science

SIMON FRASER UNIVERSITY

Spring 2006

APPROVAL

Name:	**Wei Shen**
Degree:	**Master of Science**
Title of Thesis:	**BGP Route Flap Damping Algorithms**

Examining Committee:

Chair: **Dr. Rob Cameron**
Professor of the School of Computing Science

____

Dr. Ljiljana Trajkovic
Senior Supervisor
Professor of the School of Engineering Science

____

Dr. Uwe Glässer
Supervisor
Associate Professor of the School of Computing Science

____

Dr. Qianping Gu
Examiner
Professor of the School of Computing Science

Date Defended/Approved: ____

ABSTRACT

Route flap damping (RFD) is a mechanism used in Border Gateway Protocol (BGP) to prevent persistent routing oscillations in the Internet. It plays an important role in maintaining the stability of the Internet routing system. RFD works by suppressing routes that flap persistently. Several existing algorithms address the issue of identifying and penalizing route flaps. In this thesis, we compare three such algorithms: *original RFD, selective RFD,* and *RFD+*. We implement these algorithms in ns-2 and evaluate their performance. We also propose two possible improvements to the RFD algorithms.

DEDICATION

To my wife Yang, my mom and dad.

ACKNOWLEDGEMENTS

I would like to give special thanks to Dr. Ljiljana Trajkovic, my senior supervisor, for her guidance, support, and encouragement during my studies at Simon Fraser University. Thank you for your patient instructions and valuable ideas for my research and academic writing.

My gratitude also goes to Dr. Qianping Gu and Dr. Uwe Glässer for serving on my examining committee. Thank you for your precious time in reviewing my thesis and providing insightful comments and suggestions.

I would like to express my sincere thanks to the people who helped me throughout this project. Without you, this thesis would not have been completed. Many thanks to Tony Dongliang Feng for the implementation of ns-BGP 2.0 and to the authors of SSFNet BGP-4 v1.5.0, Brian J. Premore and Zhuoqing Morley Mao, for their wonderful implementation of BGP. Thanks also go to my fellow graduate students in the Communication Networks Laboratory: Nenad Laskovic, Nikola Cackov, Jiaqing James Song, Andre Dufour, Renju Narayanan, Bozidar Vujicic, Svetlana Vujicic, Savio Lau, Wan Gang Zeng, Hao Johnson Chen, Hao Leo Chen, Qing Kenny Shao, and Grace Hui Zhang, for their support and friendship.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

AS	Autonomous System
BGP	Border Gateway Protocol
BGP-4	Border Gateway Protocol version 4
BRITE	Boston University Representative Internet Topology Generator
DOS	denial of service
GLP	Generalized Linear Preference
ICMP	Internet Control Message Protocol
IP	Internet Protocol
MED	multi-exit discriminator
MRAI	Minimum Route Advertisement Interval
RFC	Request for Comments
RFD	Route Flap Damping
RIPE	Réseaux IP Européens
SSFNet	Scalable Simulation Framework Network models
TCP	Transmission Control Protocol

1 INTRODUCTION

The Border Gateway Protocol (BGP) [1], [2] is an inter-Autonomous System (AS) routing protocol. It defines the way inter-domain routers in the Internet communicate with each other. The main function of a BGP speaking system (BGP speaker) is to exchange network reachability information with other BGP speakers. BGP-4 is the current de facto exterior routing protocol in the Internet.

1.1 BGP and Network Instability

BGP is a path vector routing protocol. It limits the distribution of a router's reachability information to its neighbor routers (peers). This information includes a list of AS numbers that a route traverses from source to destination. It is used to construct an AS connectivity graph so that routing loops may be identified and removed and that AS-level policies may be executed. BGP has the characteristic of being incremental, which implies that updates (advertisements or withdrawals) are sent only when routing tables change. When a BGP speaker receives new updates from one of its peers, it will apply a decision process to select the best route and to update its own routing table if necessary. It will also send update messages to other peers to notify them of the changes in its routing table [3]-[8]. This property of BGP suggests that the change of routing information in one BGP speaker has a direct impact on the number of BGP update messages generated within the network and the processing load imposed on other BGP speakers. In an ideal situation, routers may only need to generate updates for the relatively infrequent changes in the routing policy or the network topology such as the addition of new (or removal of existing) physical networks. However, problems such as router configuration errors, transient physical and data link failures, software defects, and insufficient computer processing power occur in the Internet, resulting in the rapid change of network reachability and topology information [9]–[13]. Network instability may greatly increase the generated BGP updates and may lead to poor network performance.

1.2 Route Flap Damping in BGP

Route flap damping (RFD) [14] is a mechanism used in BGP to control the frequency of route updates caused by possible massive changes in the network routing state. The phenomenon that a route oscillates between being available and unavailable is known as *route flapping* [15]. A typical example of a route flap is that a route is first advertised, then withdrawn, and then re-advertised. By suppressing routes that persistently flap and stopping them from being further advertised, the route flap damping mechanism aims to reduce the number of BGP update messages sent within the network and to decrease the processing load imposed on BGP speakers. Well-designed route flap damping algorithms should not significantly delay the convergence of generally well-behaved routes [14]. Route flap damping is widely used to help control the propagation of severe instabilities in the Internet and to make such instabilities more localized. It plays an important role in maintaining the stability of the Internet routing system.

Several route flap damping algorithms have been proposed to identify and penalize route flaps. Three such algorithms are: o*riginal RFD* [14], *selective RFD* [16], and *RFD+* [17]. They differ in the way they identify route flaps.

1.3 Contributions

The goal of this thesis is to compare three existing RFD algorithms, evaluate their performance, and propose possible improvements. A summary of the contributions [18] follows:

1.3.1 Implementation of RFD Algorithms in ns-2

We implemented the route flap damping mechanism in an existing BGP model *ns-BGP* [19]-[21] that has been developed for the network simulator ns-2 [22]. We ported relevant code from the BGP-4 implementation in the SSFNet simulation tool [23] and made necessary modifications and additions. Three existing RFD algorithms and two proposed improvements are implemented in the *ns-BGP* module. The modified *ns-BGP* [24] is a free public BGP module that implements *original RFD, selective RFD*, *RFD+*, *modified RFD+*, and *combined RFD*.

1.3.2 Performance Analysis of Three Existing RFD Algorithms

We compared the *original RFD*, *selective RFD*, and *RFD+* algorithms and evaluated their performance in several network topologies. We used a network topology generator and information from genuine BGP routing tables to build considerably larger and more realistic network topologies than those used previously [16], [17]. Using various simulation scenarios, we compared the performance of the three RFD algorithms.

1.3.3 Possible Improvements of RFD Algorithms

We suggested two possible improvements to the RFD algorithms: *modified RFD+* and *combined RFD*. *Modified RFD+* is a simple modification to *RFD+* that can identify genuine route flaps better than other RFD algorithms. *Combined RFD* is a simple adaptive approach that takes advantage of the strengths of *modified RFD+* and *original RFD*. It efficiently suppresses persistent flaps without suppressing routes that flap only once or twice.

The major contribution of the thesis is the proposal of the two improvements to the RFD algorithms: *modified RFD+* and *combined RFD*. Based on the performance analysis of three existing RFD algorithms, the proposed improvements address the issues of proper identification of route flaps and efficient reduction of update messages in cases of both occasional and persistent flaps.

1.4 Thesis Organization

This thesis is organized as follows: In Section 2, we provide background on BGP and RFD algorithms. The implementation and simulation of route flap damping in ns-2 are described in Section 3. In Section 4 and Section 5, we present performance analysis of RFD algorithms in the case of occasional and persistent flaps. Possible improvements to RFD algorithms are discussed in Section 6. We conclude with Section 7. The tcl scripts for validation tests in ns-2 are listed in Appendix A. Appendix B and Appendix C contain figures and tables that are supplementary to those included in Section 4 and Section 5 for all simulated network topologies.

2 BACKGROUND

Routers are dedicated devices that move packets of data between hosts in the Internet. They build routing tables and use the routing information in the routing tables to determine the best paths to certain destinations. Routing is performed at two levels: intra-domain and inter-domain. BGP is currently the de facto inter-domain routing protocol in the Internet. In this section, we introduce the background information regarding BGP and three existing BGP route flap damping algorithms.

2.1 BGP

The Internet consists of a large number of interconnected Autonomous Systems (ASs) loosely defined as a set of routers and networks under the same administrative control. Each AS is represented by a unique 16-bit number that is assigned by the numbering authorities such as RIPE NCC (Réseaux IP Européens Network Coordination Center). Routers in different ASs use BGP to exchange network reachability information with each other, as illustrated in Figure 2.1.

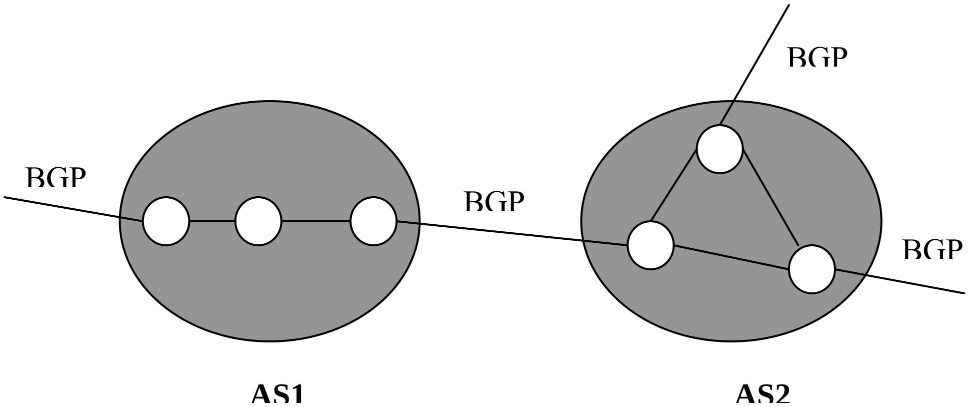

Figure 2.1: BGP: an inter-domain routing protocol.

2.1.1 Peer Session Management

BGP uses the Transmission Control Protocol (TCP) as its transport protocol. This guarantees transport reliability and rules out the need for BGP to deal with retransmission, acknowledgement, and sequencing. A BGP speaker first opens a TCP connection to a neighboring BGP speaker (peer) in order to exchange routing information with each other. Once the connection is established, each side sends an *open* message to negotiate certain parameters for the current session, such as *hold time* that specifies the maximum time a BGP speaker has to wait between the receipt of successive messages. A BGP speaker also sends *keep-alive* messages regularly to its peers to monitor the reachability of the peers and to prevent their hold timers from expiring. If a BGP speaker detects an error condition such as a malformed message or if it fails to receive any message during an interval longer than the hold time, it will send a *notification* message and close the connection [1], [6], [21].

2.1.2 Exchange of Routing Information

BGP speakers use *update* messages to exchange routing information between each other. This exchange of routing information may occur between BGP speakers in distinct ASs (external BGP) or within the same AS (internal BGP). After two BGP speakers establish a peer session, they initially exchange their entire BGP routing tables. Subsequently, a BGP speaker only sends incremental updates to its peers to notify them of any changes in its routing table.

An update message is used to advertise a feasible route to a peer or to withdraw existing unfeasible routes from service. A route contains attributes of a path to a particular destination. The destination is represented by an IP address block called *prefix*, which consists of a 32-bit address and a mask length indicating the size of the network (e.g., 192.168.1.0/24). The path attributes that an update message may contain to specify the property of a route include *AS path*, *origin*, *next hop*, *local preference*, and *multi-exit discriminator* (MED). The *AS path* attribute specifies a list of AS numbers that an update has traversed from source to destination. It is used in BGP loop detection and path selection. The *origin* attribute identifies how the origin AS learned about the route. The *next hop* attribute defines the IP address of the router that should be used as the next hop to a particular destination. The *local preference* attribute indicates the preferred path when multiple paths to the same destination exist. It is exchanged among routers within the same AS. The *multi-exit discriminator* attribute hints to the peers in a neighboring AS about the preferred path when multiple entry points to that AS exist [8].

2.1.3 Processing of Routes

When a BGP speaker receives multiple routes to a certain destination from its peers, it first applies its import policies to remove unwanted routes. For example, a BGP speaker may not accept routes advertised from a particular AS. The BGP speaker then invokes a decision process to select a single best route to the destination. During the decision process, BGP computes and compares the route preference of each feasible route to the destination. A route with the highest degree of preference (indicated by an integer) is chosen as the best route. A sequence of steps may be involved in order to choose the best route from a set of candidate routes. For example, BGP may need to compare the path attributes of two routes in the order of *local preference*, *AS path*, *origin*, and *MED* to select a preferred route [8]. The best route is placed in the BGP speaker’s routing table. Those unselected routes are stored as backups because they may be used afterwards when a route is withdrawn. Finally, the BGP speaker applies its export policies to determine whether the route should be advertised to the peers. The BGP speaker needs to modify some path attributes if it is to advertise the route [4], [21].

2.1.4 Minimum Route Advertisement Interval and Convergence Time

BGP constrains the level of routing traffic (update messages) by setting the *Minimum Route Advertisement Interval* (MRAI) parameter. MRAI defines the minimum length of time that must elapse between successive advertisements of routes to a certain destination from a single BGP speaker [1].

When a topological or policy change occurs in the Internet, BGP speakers may explore a number of transient routes before converging to a new stable route. The BGP convergence time is the time difference between the instances when the origin router sent its update message and when the last update message that resulted from the original update has been processed [25], [26]. Description of a methodology for measuring BGP pass-through time in deployed networks is given in [27].

The duration of MRAI has a significant impact on BGP convergence time. The default MRAI value (30 s) is not optimal for all types of network topologies and traffic loads. An adaptive MRAI algorithm has been proposed [28] to reduce the BGP convergence time through the adaptive adjustment of MRAI values and the adoption of reusable MRAI timers. It was shown that the adaptive MRAI algorithm may lead to a shorter convergence time while maintaining the number of update messages comparable to the current BGP without adaptive MRAI.

2.2 Route Flap Damping Algorithms

BGP employs the route flap damping mechanism to prevent the spread of persistent routing oscillations in a network. Route flap damping suppresses routes that flap persistently. A common approach in route flap damping is to assign a penalty to a route and then increment the penalty value when the route flaps. When the penalty of a route exceeds a certain threshold, the route is suppressed. A suppressed route does not participate in the BGP decision process and, hence, is not further advertised. The penalty of a route also decays exponentially over time. Once the penalty decreases below a predefined threshold, the route is reused in the decision process and may be advertised again.

The key parameters used in route flap damping are:

Penalty: a metric that is incremented each time a route flaps (different types of flaps may incur different penalty values)

Half life: a parameter that defines the time duration for the penalty to be reduced by half

Suppress limit: threshold above which a route is suppressed

Reuse limit: threshold below which a route is reused in the decision process

Maximum suppression time: maximum amount of time a route may remain suppressed, independent of the penalty

The default Cisco settings for these route flap damping parameters are listed in Table 2.1:

Table 2.1: Default Cisco settings for route flap damping.

Suppress limit	2,000
Reuse limit	750
Half life (s)	900
Withdrawal penalty	1,000
Attribute change penalty	500
Re-advertisement penalty	0
Maximum suppression time (s)	3,600

Based on the default Cisco settings, the change of route penalty over time is illustrated in Figure 2.2. The route penalty decays according to:

$$\text{penalty}(t2) = \text{penalty}(t1) * e^{(-(t2 - t1) * \ln 2 / \text{half-life})}, \qquad (1)$$

where penalty(t1) and penalty(t2) are the penalty values at times t1 and t2, respectively (t2 > t1).

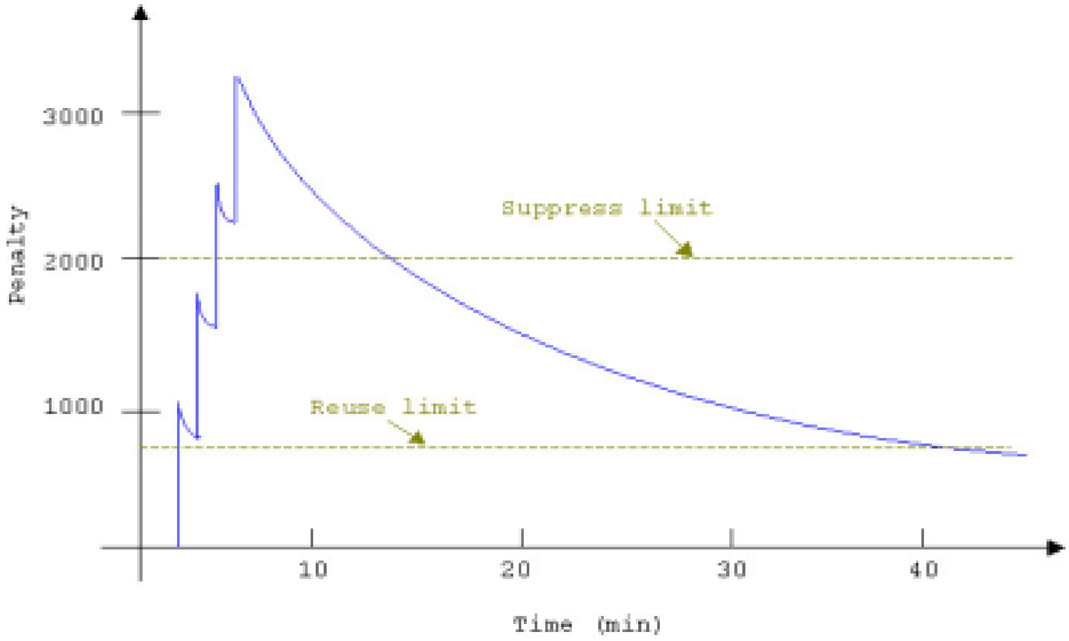

Figure 2.2: Route penalty versus time (based on default Cisco settings).

Route flap damping algorithms address the issue of identifying and penalizing route flaps. Three such algorithms are: *original RFD*, *selective RFD*, and *RFD+*. They differ in the way of identifying route flaps.

2.2.1 The Original Route Flap Damping Algorithm (*Original RFD*)

The *original RFD* algorithm, defined in RFC 2439 [14], considers each route withdrawal or route attribute change (route replacement) as a flap and penalizes it accordingly. The pseudo code for *original RFD* is shown in Algorithm 2.1. *Original RFD* is the RFD algorithm that is currently in use in the Internet. An AS administrator has the option to turn on/off the RFD mechanism implemented in a router.

```
when receiving a route $r$ with prefix $d$ from peer $j$
if (W($r$) and !W($p$))
                                        // W($x$) returns true only if $x$ is a withdrawn route
                                        // $p$ is the previous route with prefix $d$ from peer $j$
        a flap is identified: route withdrawal
else if (!W($r$) and !W($p$) and $r \neq p$)
        a flap is identified: route attribute change
$p = r$
```

Algorithm 2.1: Pseudo code of the *original RFD* algorithm.

It was shown [16], [29] that *original RFD* could significantly delay the convergence of relatively well-behaved routes (routes that flap only occasionally). This behavior arises from the interaction between route flap damping and BGP path exploration during a route withdrawal. When a route is withdrawn, BGP searches for feasible alternatives leading to the desired destination. In the case when a particular destination becomes unreachable due to a link failure, a BGP speaker tries other feasible routes to the destination until it finds no alternatives. This path exploration process may lead to penalty increase due to the interim updates because these interim updates may be falsely identified as route flaps. While a route advertised by the origin router flaps only once (one genuine route flap), the RFD mechanism in other routers may report multiple route flaps. As a result, a single route withdrawal may cause route suppressions and significantly delay BGP convergence.

2.2.2 The Selective Route Flap Damping Algorithm (*Selective RFD*)

A new algorithm called *selective RFD* was proposed [16] to distinguish BGP path explorations from genuine route flaps. It aims to correctly suppress routes that flap persistently and to remain insensitive to the interim updates during a BGP path exploration. It was observed that selection of routes during path exploration was based on the local preference in a non-increasing order. *Selective RFD* specifies that the sender attaches its local preference to each route advertisement. A flap is identified and the penalty value is incremented accordingly if the receiver detects a change of direction in route preference. An example is an increase in the route preference following a decrease. The pseudo code for the *selective RFD* algorithm is shown in Algorithm 2.2. Simulations of small networks [16] indicated that *selective RFD* identifies genuine route flaps better than *original RFD*.

```
when receiving a route r with prefix d from peer j
if (W(r) and !W(p))
                          // W(x) returns true only if x is a withdrawn route
                          // p is the previous route with prefix d from peer j
    tmp = 1               // this is a potential flap: route withdrawal
    temporarily ignore the withdrawal and remember the potential route penalty
else
    if (!W(r) and !W(p) and dop(r) > dop(p))
                          // dop(x) returns the degree of preference of route x
        curBit = 1        // comparison result is stored in curBit (for current round of
                          // comparison) and preBit (for previous round of comparison)
                          // 1: current route has a higher preference than previous one
                          // -1: current route has a lower preference than previous one
        if (preBit == −1)
            a flap is identified: route attribute change
            if (tmp == 1)
                count the temporarily ignored withdrawal as a flap
    else if (!W(r) and !W(p) and dop(r) < dop(p))
        curBit = −1
        if (preBit == 1)
            a flap is identified: route attribute change
            if (tmp == 1)
```

```
                count the temporarily ignored withdrawal as a flap
        $p = r$
        $preBit = curBit$
        $tmp = 0$
```

Algorithm 2.2: Pseudo code of the *selective RFD* algorithm.

S*elective RFD* does not always identify flaps correctly because it incorrectly assumes monotonic changes in route preference [17]. When a current best route is withdrawn, a BGP speaker selects a new best route from the set of currently feasible alternatives. However, this set of currently feasible alternatives changes over time. Better paths may become available afterwards during BGP path exploration due to topological dependencies and delays in BGP message processing and propagation. This results in non-monotonicity of route preference during path exploration.

2.2.3 The RFD+ Algorithm (*RFD+)*

The *RFD+* algorithm was proposed [17] to solve the problem that *selective RFD* posed. *RFD+* correctly distinguishes between route flaps and path explorations in the case of an occasional flap.

RFD+ requires a BGP speaker to keep track of all the routes it receives from all the peers. It uses a special data structure to store the set of all routes with a particular prefix advertised from a particular peer. When a BGP speaker receives an advertisement with prefix *d* from peer *j*, it searches for the current route in the data structure *R(d, j)* that contains all routes with prefix *d* advertised from peer *j*. If the current route does not exist in the set *R(d, j)*, it inserts the route into the set *R(d, j)* and remembers the route preference. Otherwise, the preference of the current route is compared with the previous route with the same prefix *d* from the same peer *j*. If the previous route has a higher preference than the current route, the BGP speaker only needs to remember the current route preference. If, however, the current route has a higher preference than the previous route, a route flap is identified. In this case, the set of all routes with prefix *d* from peer *j* is also cleared to avoid over-counting of flaps in the subsequent cycles. The pseudo code for the *RFD+* algorithm is shown in Algorithm 2.3.

```
when receiving a route $r$ with prefix $d$ from peer $j$
 if ($r \notin R(d, j)$ )
                 //$R(d, j)$ is the set of all routes with prefix $d$ announced from peer $j$
      insert $r$ into the set $R(d, j)$
 else if ($r \in R(d, j)$ and dop($r$) > dop($p$))
                 //degree of preference of route $r$ is higher than for the previous route $p$
      a flap is identified
      clear $R(d, j)$
```

Algorithm 2.3: Pseudo code of the *RFD+* algorithm.

Hence, a new flap is identified when the following two conditions are met:

- The BGP speaker has received the current route more than once since its previous flap.
- The current route has a higher degree of preference than its previous route.

RFD+ may distinguish genuine route flaps from path explorations because it was shown [17] that routes derived from the path exploration process do not meet both of the two conditions. Simulations of *RFD+* in small networks showed that it performed better than *selective RFD* in identifying an occasional route flap [17]. *RFD+* could correctly identify genuine route flaps in the case of a single flap.

2.2.4 Simple Example

The key difference between *original RFD*, *selective RFD*, and *RFD+* is how they identify route flaps. We provide here a simple example to illustrate the difference between the three RFD algorithms in identifying route flaps. We assume that a BGP speaker has received five updates with respect to a particular prefix from a particular peer. The characteristics of these updates are shown in Table 2.2. If the number of AS hops in a route's *AS path* attribute is used as a metric for route preference (higher number of AS hops implies lower route preference), then *original RFD*, *selective RFD*, and *RFD+* would report 4, 2, and 1 flap, respectively.

Table 2.2: Characteristics of five updates.

Update receiving order	Update type	AS path
1	Advertisement	3 5 1
2	Advertisement	3 5 7 1
3	Advertisement	3 5 7 9 1
4	Withdrawal	
5	Advertisement	3 5 1

***Original RFD*:** The last four updates (*advertisement*, *advertisement*, *withdrawal*, and *advertisement*) are all considered as flaps because they are either route withdrawal or route replacement. Hence, *original RFD* would report four flaps.

***Selective RFD*:** Only the last update (*advertisement*) signals a change of direction in route preference: a decrease in route preference followed by an increase. The fourth update (*withdrawal*) would also be considered as a flap after the last update is processed. Hence, *selective RFD* would report two flaps.

***RFD+*:** Only the last update (*advertisement*) meets two conditions: the current route has a higher preference than the previous route and the current route has appeared more than once since the previous flap. Only the last update is identified as a flap. Hence, *RFD+* would report only one flap.

3 RFD IMPLEMENATION AND SIMULATION IN ns-2

We implemented various RFD algorithms in the network simulator ns-2 [22]. We also simulated various scenarios in ns-2 for performance analysis of the RFD algorithms.

3.1 ns-2 Implementation of the RFD Algorithms

The implementation of various RFD algorithms [24] is based on the ns-BGP module [19] developed for ns-2 and the SSF.OS.BGP4 module from SSFNet [23]. The ns-BGP module provides an underlying BGP implementation for the route flap damping mechanism. The *original RFD* and *selective RFD* algorithms were already implemented in SSFNet BGP-4 v1.5.0. We ported the relevant code from SSFNet to ns-BGP and made necessary modifications. We added the implementation of other RFD algorithms to ns-BGP.

3.1.1 ns-BGP Routing Structure

The routing structure of the modified ns-BGP module [20], [21] is shown in Figure 3.1. It consists of the forwarding plane (classifying and forwarding packets to their destinations) and the control plane (computing routes, maintaining routing tables, and implementing routing algorithms) [30]. *Classifier* and *routing module* belong to the forwarding plane while *route logic*, *route object*, and *routing protocol* are part of the control plane. The shaded areas in the figure are the important C++ classes used to implement BGP with route flap damping mechanism. The two key C++ classes specific to route flap damping in the modified ns-BGP are: *DampInfo* and *ReuseTimer* (darker

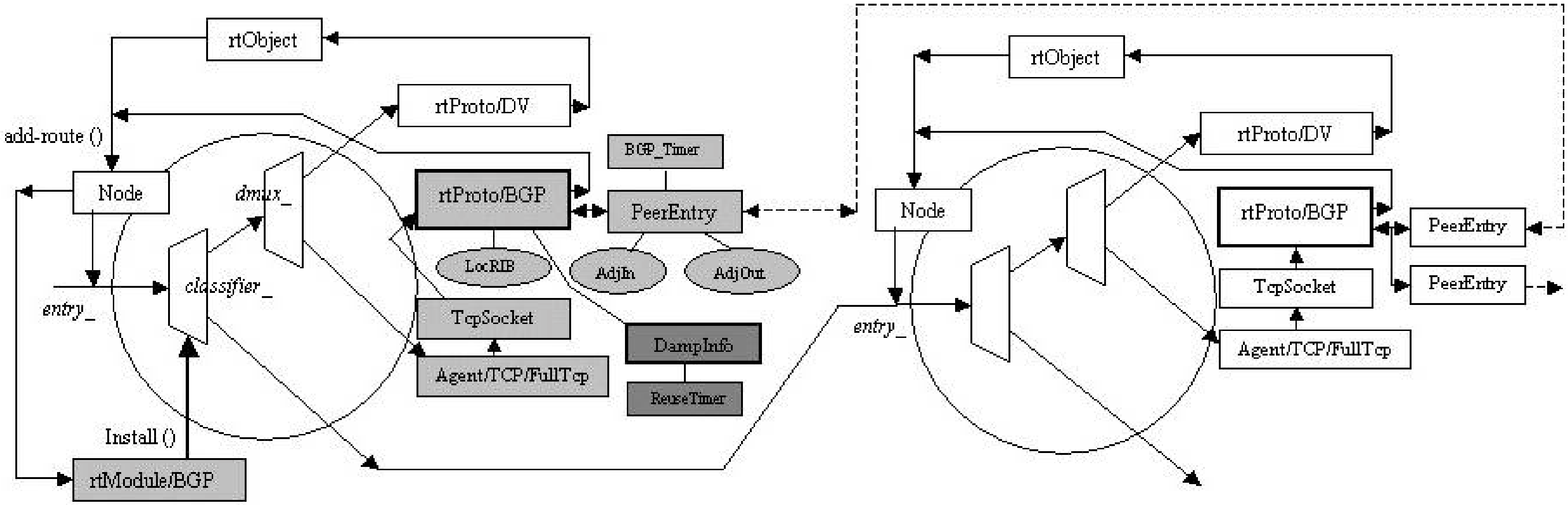

Figure 3.1: Routing structure of ns-BGP with route flap damping mechanism.

shade). The *DampInfo* class stores the damping structure for a prefix advertised from a peer of a BGP speaker. This class also implements the various RFD algorithms. The *ReuseTimer* class keeps track of the reuse timer associated with a suppressed route. When a reuse timer expires, the suppressed route is reused in the BGP decision process and may be advertised again. *VecRoutes* is another class used for route flap damping. It maintains an array of the interim routes in the *RFD+* algorithm. We modified two existing C++ classes in the original ns-BGP, *RouteInfo* and *rtProtoBGP*, to implement the route flap damping mechanism when update messages are received and routing decisions are made. The file *global.h*, which defines all the global variables used in BGP, was also modified to include global variables that are specific to route flap damping. These global variables, together with their default values, are also included in the file *ns-default.tcl*, a tcl script that sets the default values of all the global parameters used in ns-2.

3.1.2 Implementation Features

Route flap damping is a user-friendly feature implemented in the modified version of ns-BGP. A user may easily turn on/off route flap damping and choose a particular RFD algorithm in tcl scripts without the need to recompile the C++ source code. The same holds for the configuration of RFD parameters. RFD parameters such as *suppress limit*, *reuse limit*, *half life*, *maximum suppression time*, *withdrawal penalty*, and *attribute change penalty* may be manually configured in tcl scripts.

3.1.3 Validation Tests

We tested the implemented route flap damping mechanism with simulation scenarios that vary in network topology, simulation duration, and number of simulated flaps. The tested network topologies include a 2-node *line* topology, a 4-node *tree* topology, a 6-node *clique* topology, and an 11-node *fork* topology. The duration of the simulation varies from 250 s to 12,800 s. The number of flaps experienced in a node varies from 1 flap to 35 flaps. The tcl scripts for the tests are listed in Appendix A. The characteristics of these test scripts are summarized in Table 3.1. We examine the BGP routing table in each node and the route penalty, suppression state, and number of flaps reported by each node for every route at various moments during the simulation. The simulation results indicate that the implemented route flap damping mechanism works as expected.

Table 3.1: Characteristics of test scripts.

Topology	Duration (s)	Number of flaps	Subsection
Line	3100	4	A.1
Tree	12800	35	A.2
Clique	910	1	A.3
Fork	250	2	A.4

3.2 RFD Simulation Scenarios

Four factors are considered when designing simulation scenarios: network topology, inter-arrival time between updates, total simulation time, and the nature of flaps to simulate.

3.2.1 Network Topology

We use the BRITE (v2.1) network topology generator [31] to create network topologies that also include link delays. BRITE’s Generalized Linear Preference (GLP) model [32] is adopted to generate AS-level network topologies ranging from 100 to 500 nodes. The GLP model is “more successful than most of the other AS-level topology generators at matching the power law exponent and the clustering behavior of the Internet" [32]. The adopted values for GLP specific parameters are shown in Table 3.2.

Table 3.2: GLP specific parameters.

Node placement	Random
Growth type	Incremental
Preferential connectivity	On
p	0.45
beta	0.64
m	1

In addition to BRITE-generated topologies, we also use network topologies with 29 and 110 nodes that are built from genuine BGP routing tables. These topologies represent connected sub-graphs of the genuine AS graphs in the Internet. They were built by B. J. Premore [33] in the following way:

- generate a network topology from a BGP table dump using resources such as RouteViews
- merge nodes together by first choosing nodes with the smallest degrees until the topology is reduced to 1,000 nodes
- prune a given percentage of links from the topology and keep the largest remaining connected component
- merge nodes together by choosing nodes with the smallest degrees first; do not merge if nodes share any peers and if degree of both nodes is greater than 2.

3.2.2 Inter-Arrival Time between Route Updates

One origin router is used to advertise a route to other routers in the network. All other routers respond accordingly to the messages sent by this origin router.

For each network topology, we select at least three different values for the inter-arrival time between updates sent by the origin router: a value smaller than the default Minimum Route Advertisement Interval (MRAI) of 30 s (10 s), an intermediate value (100 s), and a value large enough for BGP to converge (1,000 s).

A typical simulation scenario is:

BGP initial set-up

Origin router Z advertises a route R at the n-th second

Origin router Z withdraws the route R at the (n+i)th second

Origin router Z re-advertises the route R at the (n+2i)th second

...

Simulation terminates at the m-th second,

where *i* is the inter-arrival time between updates, and the ellipsis '…' indicates there may be multiple cycles of withdrawals and re-advertisements, depending on the nature of flaps we wish to simulate.

3.2.3 Scenario Simulation Time

The duration of a simulation depends on whether or not the scenario contains the route suppression period when BGP nodes wait for routes to become reused and advertised again. By comparing scenarios with and without the suppression period, we can evaluate the impact of route suppression on individual BGP speakers and on the network. In the case of occasional flaps, the typical duration of simulation with and without the route suppression period is ~2,500 s and ~6,000 s, respectively.

3.2.4 Nature of Flaps to Simulate

To test the effectiveness of the route flap damping algorithms, we evaluate their performance in cases of occasional and persistent flaps. Occasional flaps are route flaps that occur sporadically, while persistent flaps are route flaps that occur at frequent intervals. RFC 2439 [14] does not provide precise definitions of occasional and persistent flaps. We mimic occasional and persistent flaps by using one and five flaps within a certain period of time, respectively. We opt for a small number of flaps in both cases because it simplifies the simulation process without defeating the purpose to demonstrate the nature of flaps. Note that in the case of persistent flaps, we choose number of flaps ≥ 4 because the RFD implementation requires at least four flaps for a route to be suppressed. It has been recommended [34] that suppression should not start until the fourth flap in a row. The inter-arrival times between updates are 1,000 s and 300 s for occasional and persistent flaps, respectively. The value of 1,000 s is sufficiently large for BGP to converge in the simulated networks. This enables us to adequately compare the behavior of RFD algorithms in identifying route flaps and to examine the impact of route suppression in the case of occasional flaps. A smaller value of 300 s is adopted in the case of persistent flaps. It suggests that route flaps occur at more frequent intervals. Furthermore, when the inter-arrival time between updates is relatively small (e.g., 300 s), a route that has been suppressed due to persistent flaps remains suppressed when the origin router sends its last advertisement. This suppression of the route is an indicator that the RFD algorithm can effectively suppress persistent flaps. The default MRAI value (30 s) also affects our choice of inter-arrival times between updates. Figure 3.2 shows the process for occasional and persistent flaps. Note that the order of updates matters: the sequence *advertisement*, *withdrawal*, and *advertisement* makes the route available, while the sequence *advertisement*, *advertisement*, and *withdrawal* makes the route unavailable.

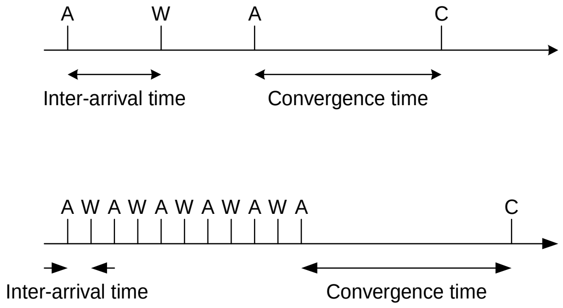

Figure 3.2: Timeline for occasional flaps (top) and persistent flaps (bottom): A (advertise), W (withdraw), and C (converge).

4 PERFORMANCE ANALYSIS OF RFD ALGORITHMS: OCCASIONAL FLAPS

Our performance analysis is focused on the effect of route flap damping and the comparison of the three existing RFD algorithms. We examine the following variables:

- total number of update messages generated by all the BGP speakers
- total number of flaps reported for the entire network
- total number of flaps reported by each BGP speaker
- maximum number of flaps associated with a single peer of each BGP speaker
- total number of route suppressions caused by all the flaps in the network
- convergence time.

We calculate the convergence times for the overall network and for individual BGP speakers. The convergence times indicate how fast the network and each BGP speaker reach a stable status regarding a particular route. The examined variables are good indications of the behavior of RFD algorithms. These variables show the impact of RFD algorithms on BGP convergence, the correctness of RFD algorithms in identifying genuine route flaps, and the effectiveness of RFD algorithms in reducing the number of update messages. We use the default MRAI value of 30 s and apply jitter to it. The default Cisco settings for RFD parameters are adopted, as shown in Table 2.1.

Simulation results are illustrated with graphs (generated by GnuPlot [35]) and tables. Simulation results for network topologies generated by BRITE and built from routing tables are given in Subsection 4.1 and Subsection 4.2, respectively.

4.1 Network Topologies Generated by BRITE

While comparing the three RFD algorithms, we also examine advertisement and withdrawal phases, effect of the inter-arrival time between updates, and impact of the origin router's location.

4.1.1 Advertisement and Withdrawal Phases

When the origin router advertises a new route or withdraws an existing route, other routers in the network will respond to this advertisement or withdrawal. BGP eventually converges when no router spends any more time processing update messages regarding the route. The advertisement phase differs significantly from the withdrawal phase in terms of convergence time and numbers of updates, flaps, and suppressions.

Figure 4.1 shows the comparison of convergence times (in seconds) for individual BGP speakers between the advertisement phase and withdrawal phase in a 100-node network when route flap damping is disabled. Approximately 75% of the BGP speakers have a considerably longer convergence time (over 100 s) during the withdrawal phase than during the advertisement phase. BGP convergence time for the network is ~200 s longer during the withdrawal phase than during the advertisement phase. These convergence times are obtained by placing the origin router at a particular location in the network. Changing the location of the origin router may lead to different convergence times. Regardless of the location of the origin router, BGP converges considerably slower during the withdrawal phase than during the advertisement phase. This is also true when route flap damping is enabled, as shown in Figures 4.2 – 4.4. The convergence times for the two phases in other networks (200, 300, 400, and 500 nodes) are shown in Figures B.2 – B.5 and Tables B.1 – B.4 in Appendix B. These figures and tables show similar results for the advertisement and withdrawal phases.

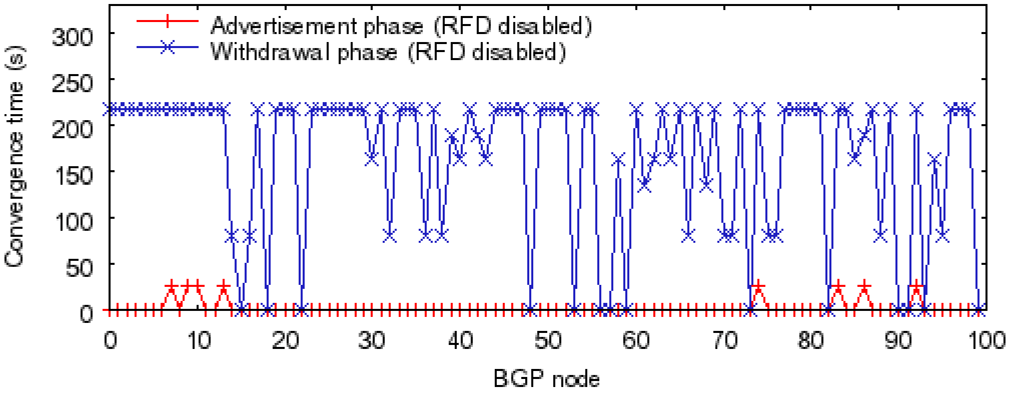

Figure 4.1: Occasional flaps in *RFD disabled*: convergence times for individual BGP nodes during the advertisement and withdrawal phases in a 100-node network.

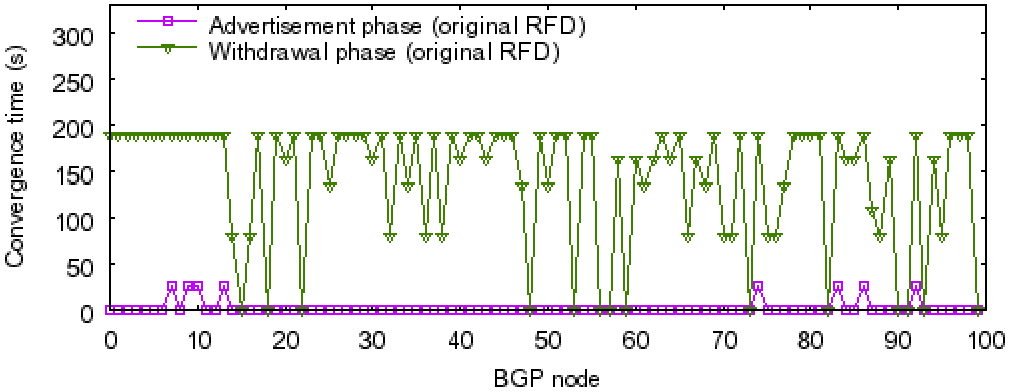

Figure 4.2: Occasional flaps in *original RFD*: convergence times for individual BGP nodes during the advertisement and withdrawal phases in a 100-node network.

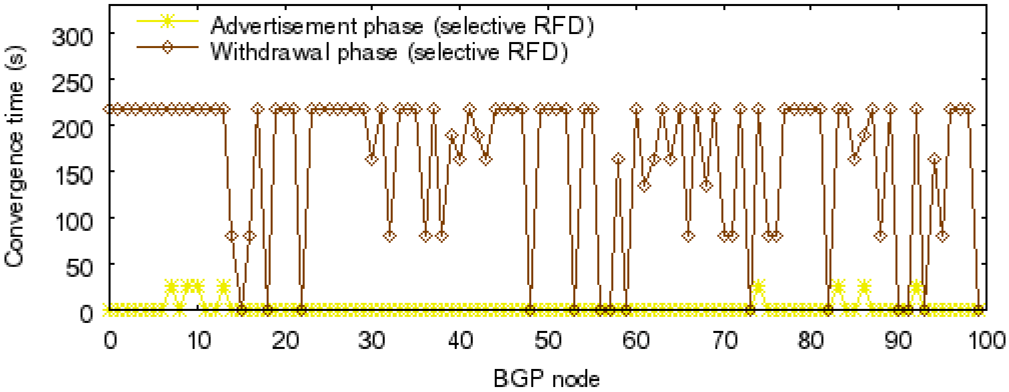

Figure 4.3: Occasional flaps in *selective RFD*: convergence times for individual BGP nodes during the advertisement and withdrawal phases in a 100-node network.

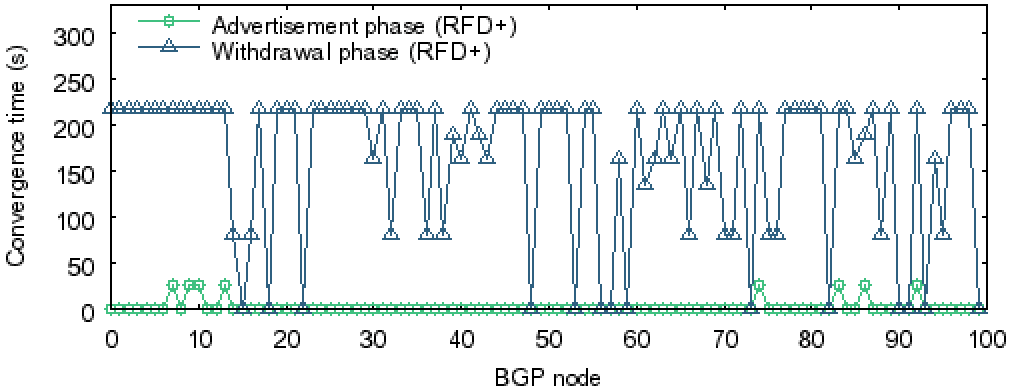

Figure 4.4: Occasional flaps in *RFD+*: convergence times for individual BGP nodes during the advertisement and withdrawal phases in a 100-node network.

The comparison between the two phases for various networks and RFD algorithms is shown in Figure 4.5 (convergence time) and Figure 4.6 (number of updates). Figure 4.5 shows that a withdrawal message takes BGP significantly longer (up to ~25 times longer) to converge than an advertisement message. The convergence time for withdrawal phase also tends to grow as the network size increases. The reason is that a BGP speaker will search for other feasible, though less preferred, routes to the desired destination when a route is withdrawn. In the case of a link failure, a BGP speaker will try all feasible routes to the destination until it finds no alternatives. This process of path exploration can be rather long, depending on the number of feasible paths that the BGP speaker has maintained for a certain prefix. As the network sizes increases, the number of feasible paths to a destination also tends to grow. This causes both the BGP convergence time and the number of generated updates to grow, as shown in Figure 4.5 and Figure 4.6. It holds for all damping algorithms during the withdrawal phase. However, the increase of network size may not affect the convergence time during the advertisement phase because this lengthy process of trying all feasible routes does not occur during the advertisement phase.

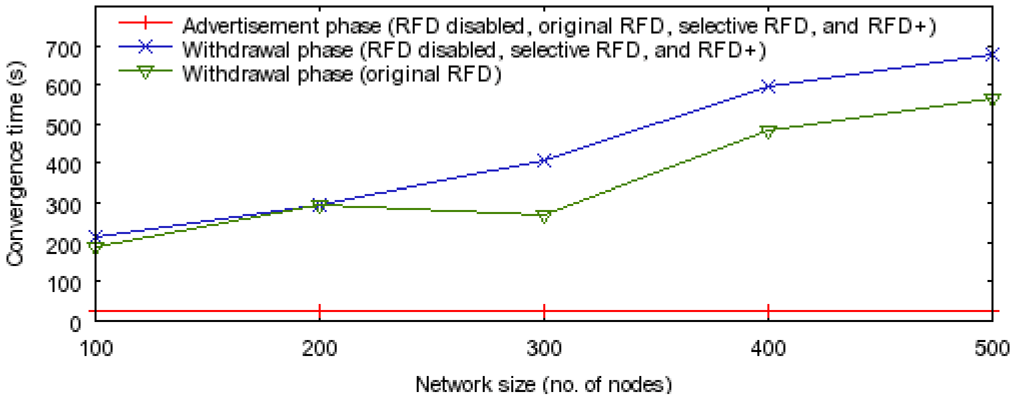

Figure 4.5: Occasional flaps: comparison of convergence time between the advertisement and withdrawal phases in various networks.

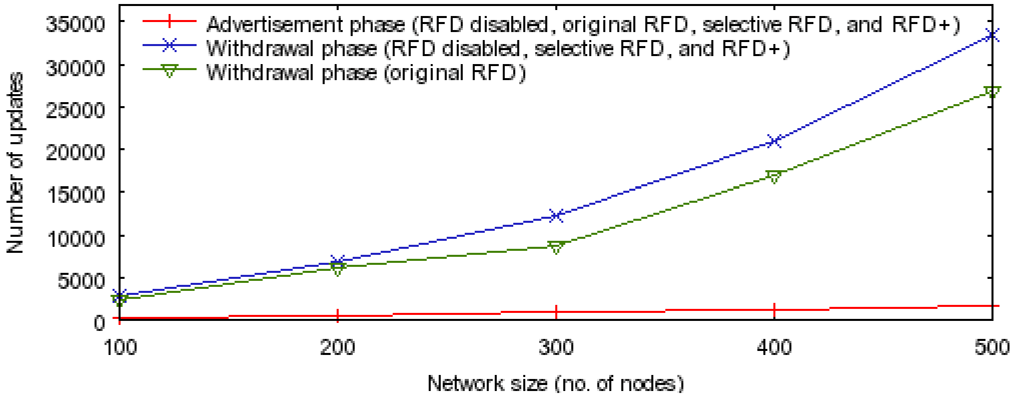

Figure 4.6: Occasional flaps: comparison of number of updates between the advertisement and withdrawal phases in various networks.

During the advertisement phase, *RFD disabled*, *original RFD*, *selective RFD*, and *RFD+* behave identically. This is because no route suppression occurs during the advertisement phase regardless of the route flap damping algorithm. This also confirms that advertisement-triggered suppression is rare [16]. However, during the withdrawal phase, even though *RFD disabled*, *selective RFD*, and *RFD+* behave identically, they differ from *original RFD*. The reason is that *selective RFD* and *RFD+* do not cause a sufficient number of route flaps in particular nodes to affect the convergence time and number of generated updates. In the case of *original RFD*, a single withdrawal message causes a large number of flaps in the network, resulting in many route suppressions that affect both the convergence time and the number of updates. Route suppression prevents some routes from being advertised subsequently and leads to fewer update messages and shorter convergence time during the withdrawal phase. For example, in the 300-node network, *original RFD* results in ~30% fewer updates (Figure 4.6) and ~35% shorter in convergence time (Figure 4.5) compared to other algorithms. Aggressive damping in *original RFD* helps BGP to converge faster after a route withdrawal. Yet, there is a side effect: in some nodes the route is suppressed and will not be advertised to other nodes even when it becomes available subsequently.

RFD algorithms play an important role during the withdrawal phase because withdrawal-triggered suppressions are common. During the withdrawal phase, *original RFD* has the fastest convergence because of its most aggressive route suppression.

4.1.2 Effect of Inter-Arrival Time between Route Updates

We examine the effect of inter-arrival time on BGP convergence time and numbers of updates, flaps, and suppressions. We calculate convergence time as the time difference between the time when the origin router re-advertises the route after withdrawal and the time when a BGP node receives its last update message. Any possible suppression period is not considered. We experiment with inter-arrival times equal to 10 s, 20 s, 30 s, 50 s, 100 s, 300 s, 500 s, 800 s, and 1,000 s in networks with 100 and 200 nodes. In networks with 300, 400, and 500 nodes, inter-arrival times are set to 10 s, 100 s, and 1,000 s only.

The effect of inter-arrival time between updates on the convergence time and numbers of updates, flaps, and route suppressions in the 200-node network is shown in

Figures 4.7 – 4.10. The results for all five networks (100 nodes – 500 nodes) are given in Tables B.5 – B.9 in Appendix B.

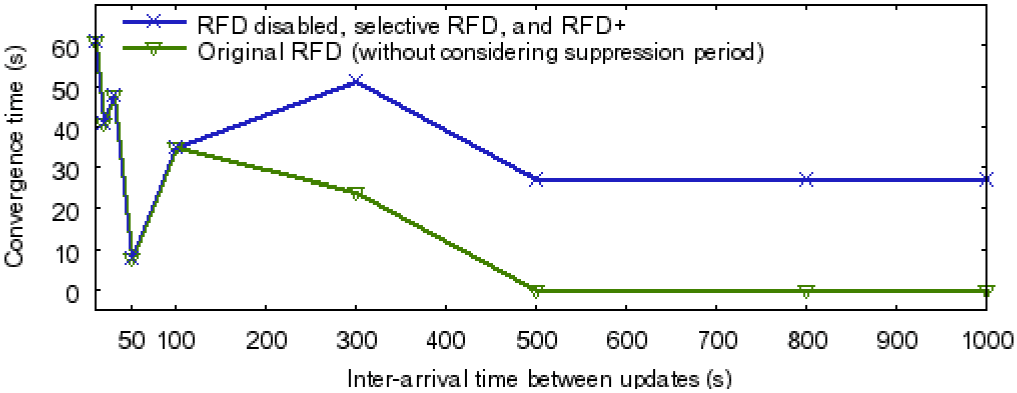

Figure 4.7: Occasional flaps: effect of inter-arrival time between updates on BGP convergence time in a 200-node network.

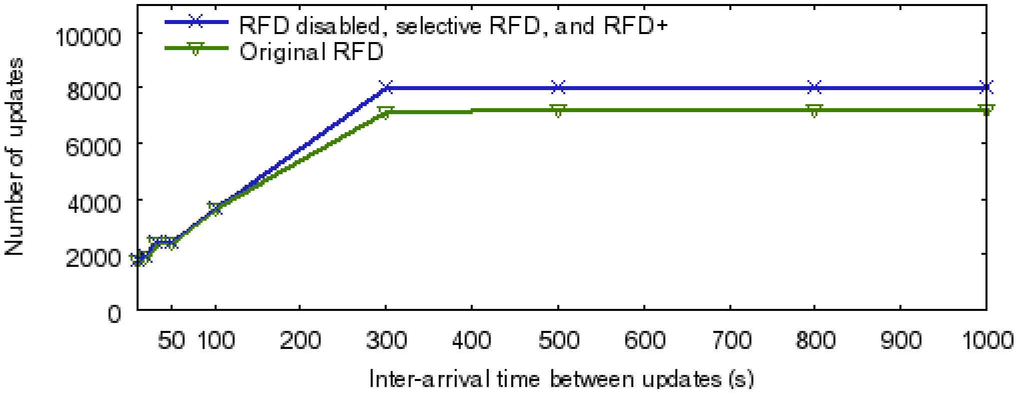

Figure 4.8: Occasional flaps: effect of inter-arrival time between updates on the number of update messages in a 200-node network.

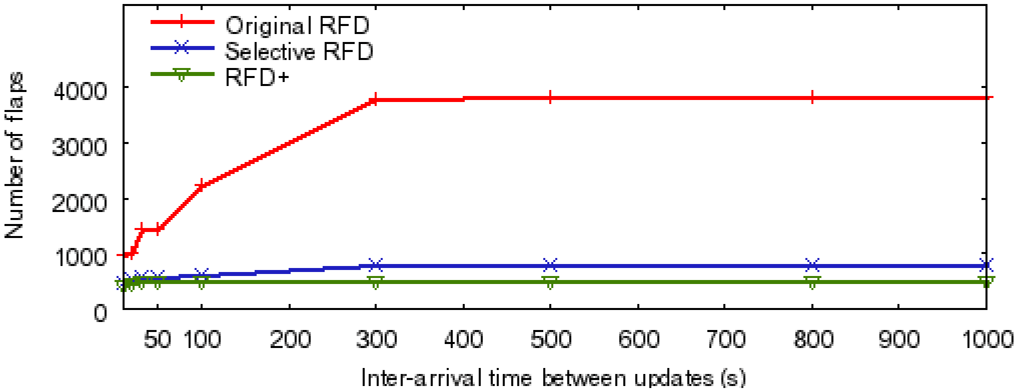

Figure 4.9: Occasional flaps: effect of inter-arrival time between updates on the number of flaps in a 200-node network.

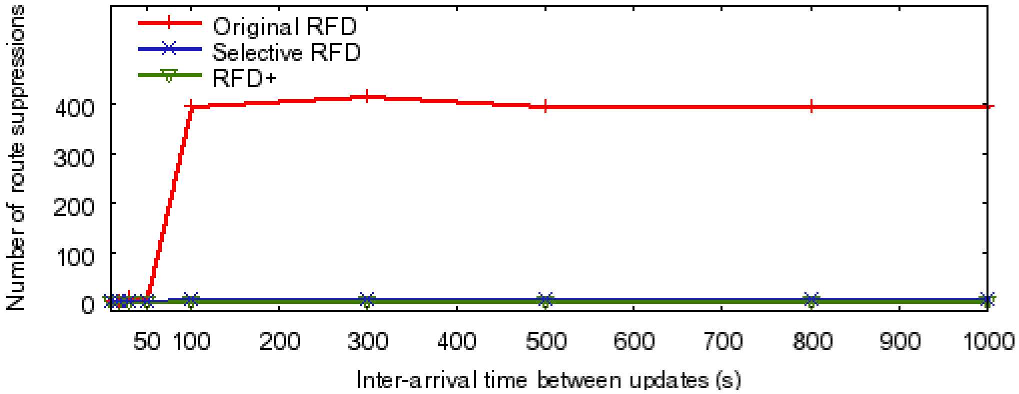

Figure 4.10: Occasional flaps: effect of inter-arrival time between updates on the number of route suppressions in a 200-node network.

Figure 4.7 and Tables B.5 – B.9 indicate that there is no visible relationship (monotonic increase or decrease) between the inter-arrival time and the convergence time. However, the number of updates and number of flaps tend to increase as the inter-arrival time increases, as shown in Figure 4.8 (number of updates) and Figure 4.9 (number of flaps). Beyond a certain threshold, further increase in inter-arrival time has no effect on either the convergence time or the number of updates, flaps, and suppressions.

When the origin router sends consecutive updates within short intervals, certain updates that wait in the queues of BGP nodes to be sent afterwards due to the MRAI constraint, may be replaced by new updates (with the same prefix to the same peer) arriving to the queues. This causes fewer update messages to be sent to the network, which, in turn, may reduce the number of flaps and the number of route suppressions. Furthermore, because BGP convergence due to a withdrawal may take a long time, the short interval between updates interrupts this lengthy process of path exploration, and prevents many updates from being generated. When the time interval increases, the number of updates tends to grow as the interruption of path exploration is delayed or may not occur at all. However, when the inter-arrival time increases beyond a certain threshold (time needed for BGP to converge), further increase of inter-arrival time does not have any effect because there are no updates generated after the threshold point.

When the inter-arrival time between updates is short (10 s – 50 s), the convergence time may be relatively long even though the total number of updates is not large. Convergence time is not proportional to the number of updates (longer convergence time does not always correspond to larger number of updates), as shown in Figures 4.7 and 4.8. This is due to the MRAI constraint that prevents one update message from being sent right after another. The difference between the instances when an update is ready to be sent and when the MRAI timer expires may strongly affect the length of convergence time. If an update is ready to be sent and the MRAI timer expires at that moment, this update message may be sent immediately. However, if an update is ready to be sent but a previous update was sent a short while earlier, the current update has to wait for up to ~30 s (the default MRAI value) before being sent. In this case, the convergence time is likely to increase. Figures 4.7 and 4.8 also indicate that in the case of an occasional flap, damping algorithms do not affect the convergence time and the number of updates when the inter-arrival time is short since all algorithms result in identical convergence time and number of updates. This is because there are no or only few route suppressions in the network.

When the inter-arrival time is large (500 s – 1,000 s), *original RFD* causes a large percentage (over 50%) of updates to be identified as route flaps. In the case of *original RFD*, there are ~3,800 flaps (Figure 4.9) among a total of ~7,000 updates (Figure 4.8) in the 200-node network. *Selective RFD* and *RFD+* behave better because they were designed to distinguish genuine route flaps from path explorations. *RFD+* performs even better than *selective RFD* in identifying route flaps. As shown in Figures 4.7 and 4.8, *RFD+* has identical convergence time and number of updates as *selective RFD*. However, *RFD+* incurs fewer route flaps and suppressions than *selective RFD*. In the 200-node network, *RFD+* reports ~40% fewer flaps than *selective RFD*, as shown in Figure 4.9. Some route suppressions occur in *selective RFD*, but none in *RFD+*, as shown in Figure 4.10 and in Table B.6 where the exact number of route suppressions is listed.

4.1.3 Location of the Origin Router

We use two different locations for the origin router for each network topology. The origin router is connected either to the core of the network (a node with the most connections) or to the edge of the network (a leaf node with only one connection). The comparison of the convergence time and number of updates between these two cases is shown in Table 4.1, when route flap damping is disabled (see Tables B.11 – B.13 in Appendix B for cases when route flap damping is enabled). The effect of the origin router's location in various networks is also illustrated in Figure 4.11 (convergence time) and Figure 4.12 (number of updates), where we consider the withdrawal phase in *original RFD*. The inter-arrival time between updates is 1,000 s.

Table 4.1: Occasional flaps in *RFD disabled*: effect of the origin router's location.

Phase	Location	Evaluation parameters	Network size (No. of nodes)				
			100	200	300	400	500
Advertisement phase	Connected to core	Convergence time (s)	27.017	27.017	27.017	27.017	27.018
		No. of updates	275	552	959	1169	1626
	Connected to edge	Convergence time (s)	27.019	54.019	27.019	81.018	54.02
		No. of updates	275	670	1011	1978	2939

Phase	Location	Evaluation parameters	Network size (No. of nodes)				
			100	200	300	400	500
Withdrawal phase	Connected to core	Convergence time (s)	216.21	297.31	405.3	594.21	675.3
		No. of updates	2857	6952	12208	21006	33390
	Connected to edge	Convergence time (s)	216.21	351.3	486.21	594.41	756.31
		No. of updates	2857	8122	15189	24527	36820

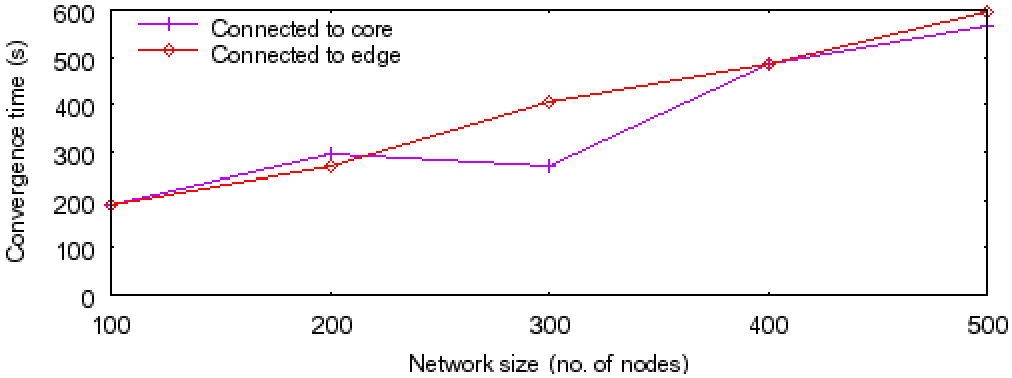

Figure 4.11: Occasional flaps in *original RFD*: effect of the origin router's location on convergence time in various networks during the withdrawal phase.

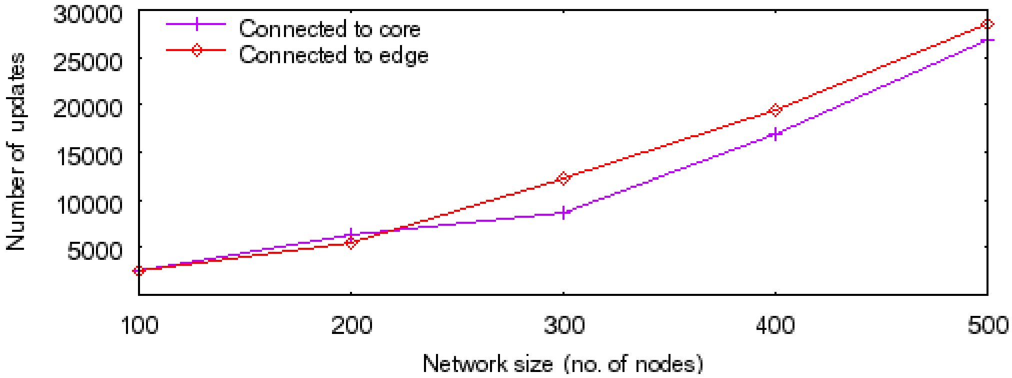

Figure 4.12: Occasional flaps in *original RFD*: effect of the origin router's location on the number of updates in various networks during the withdrawal phase.

When the sender is located at the edge of the network, it often takes BGP up to ~20% longer to converge than when the sender is located at the core of the network. This difference in convergence time may increase significantly for certain network topologies during the advertisement phase. For example, in the 400-node network, the convergence time during the advertisement phase increases by ~200% (from 27.017 s to 81.018 s), as shown in Table 4.1. The total number of generated updates also tends to increase when the sender is located at the edge of the network. Positioning the origin router at the edge of the network often results in up to ~25% increase in the number of updates, compared to placing the origin router at the core of the network. In the 500-node network, this difference in the number of updates during the advertisement phase increases to ~80% (from 1,626 to 2,939), as shown in Table 4.1. An exception is the withdrawal phase in the 200-node network with *original RFD* (Figures 4.11 and 4.12): the convergence time and the number of updates decrease when the origin router is located at the edge of the network. This suggests that, in spite of the general trend, the effect of the origin router's location on the convergence time and the number of updates depends on the network topology, the phase (advertisement or withdrawal), and the damping algorithm.

4.1.4 Impact of Route Suppression

Route flap damping causes a route to be suppressed. Consequently, some BGP speakers may not be able to receive the route re-advertisement until their peer(s) reuse the route and advertise it again. The suppression time ranges from ~20 min to 1 h. The comparison of the convergence time and numbers of updates, flaps, and suppressions between the case when route suppression period is not taken into account and the case when route suppression period is considered is shown in Table 4.2. The convergence times for individual BGP nodes in the two cases are shown in Figure 4.13 (a 200-node network) and Figure 4.14 (a 500-node network). These figures illustrate the comparison between the two cases only when *original RFD* is enabled, because *selective RFD* and *RFD+* do not cause the suppression period when an occasional flap occurs. The comparison enables us to evaluate the impact of route suppression on individual BGP speakers and on the overall network.

Table 4.2: Occasional flaps: impact of route suppression on network performance for various RFD algorithms.

Algorithm	Condition	Evaluation parameters	Network size (no. of nodes)				
			100	200	300	400	500
Original RFD	Not considering suppression period	Convergence time (s)	0.02	0.02	0.02	27.01	0.02
		No. of updates	2971	7202	10371	18979	29662
		No. of flaps	1502	3829	6196	11181	17877
		No. of suppressions	175	395	658	880	1194
	Considering suppression period	Convergence time (s)	1560.1	2071.7	2134.9	3430.0	3472.7
		No. of updates	2984	7380	10464	19179	29924
		No. of flaps	1509	3964	6257	11317	18053
		No. of suppressions	175	395	658	880	1194
Selective RFD	Not considering suppression period	Convergence time (s)	27.02	27.02	27.02	27.02	27.02
		No. of updates	3407	8056	14126	23344	36642
		No. of flaps	334	802	1166	1590	1880
		No. of suppressions	0	3	6	13	6
	Considering suppression period	Convergence time (s)	27.02	27.02	27.02	27.02	27.02
		No. of updates	3407	8056	14126	23344	36642
		No. of flaps	334	802	1166	1590	1880
		No. of suppressions	0	3	6	13	6
RFD+	Not considering suppression period	Convergence time (s)	27.02	27.02	27.02	27.02	27.02
		No. of updates	3407	8056	14126	23344	36642
		No. of flaps	261	497	813	1049	1386
		No. of suppressions	0	0	0	0	0
	Considering suppression period	Convergence time (s)	27.02	27.02	27.02	27.02	27.02
		No. of updates	3407	8056	14126	23344	36642
		No. of flaps	261	497	813	1049	1386
		No. of suppressions	0	0	0	0	0

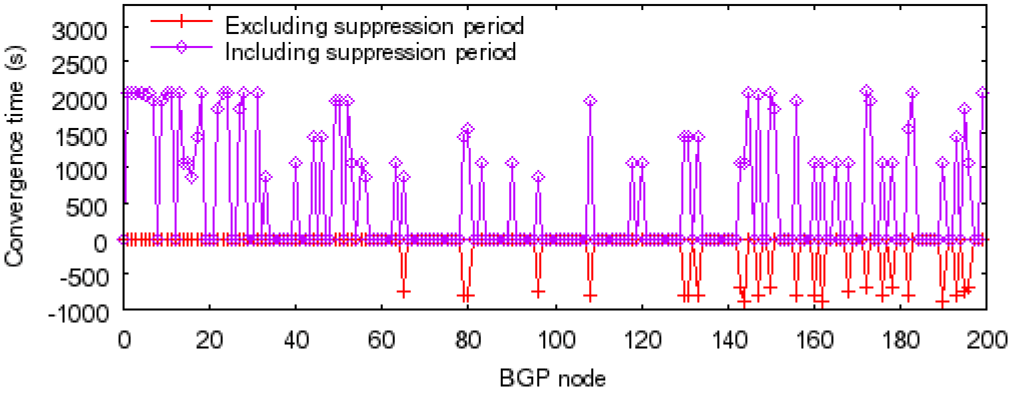

Figure 4.13: Occasional flaps in *original RFD*: impact of route suppression on the convergence time of BGP nodes in a 200-node network.

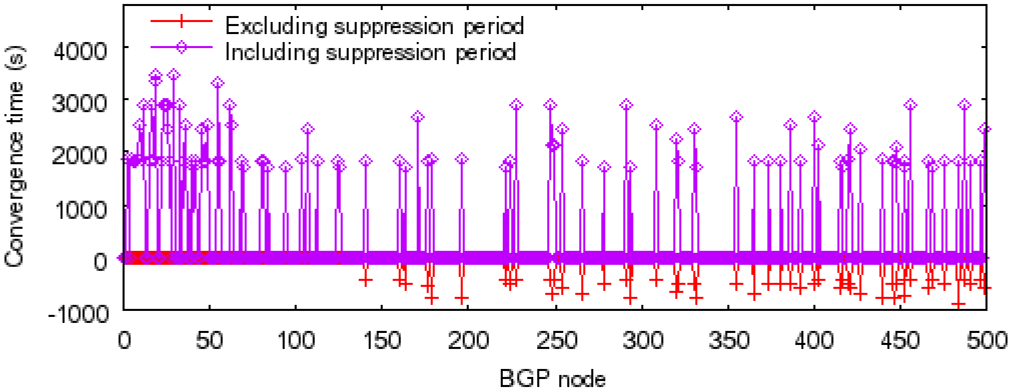

Figure 4.14: Occasional flaps in *original RFD*: impact of route suppression on the convergence time of BGP nodes in a 500-node network.

Figure 4.13 and Figure 4.14 illustrate the effect of one withdrawal message (one flap) on network performance, such as the number of BGP nodes that suffer from delayed convergence (nodes that have convergence time > 800 s) due to route suppression.

In the case of *original RFD*, a single flap may cause a large number of route suppressions in the network, as indicated by points in the negative range in Figures 4.13 and 4.14. The negative values are obtained by calculating the time difference between the instance when the origin router re-advertises the route and the instance when a BGP node receives its last update message regarding the route, as shown in Figure 4.15. The last update (*U*) occurs after route re-advertisement (second *A*) in the case when route suppression period is taken into account. However, in the case when route suppression period is not considered, the last update (*U*) occurs before route re-advertisement (second *A*). The negative values imply that nodes do not receive the route re-advertisement after withdrawal and will have to wait until other nodes become reused and start to advertise again. Negative values are shown to indicate the number of such nodes. Because *original RFD* suppresses flapping routes aggressively, when the route is re-advertised after withdrawal some nodes may not receive any new updates regarding the route (which makes the particular destination unreachable) or may receive only partial information concerning the particular destination before the route becomes reused and advertised again. Figure 4.13 shows that in the 200-node network, 12% of nodes (points in the negative range) do not receive new updates when the route is re-advertised because of route suppression. In the 500-node network, the percentage of such nodes is ~10%, as shown in Figure 4.14. Consequently, in the 200-node (500-node) network, ~30% (~20%) of the BGP nodes suffer from a long convergence delay up to ~2,000 s (~3,500 s), as indicated by points with convergence time > 800 s in Figure 4.13 (Figure 4.14). The effect of route suppression on the convergence times of individual BGP nodes in other networks (100, 300, and 400 nodes) is illustrated in Figures B.6, B.8, and B.9 in Appendix B. They show results that are similar to Figures 4.13 and 4.14.

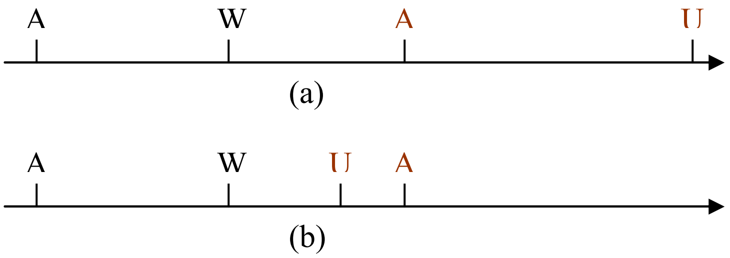

Figure 4.15: Comparison of the order of update messages between (a) when route suppression period is taken into account and (b) when route suppression period is not considered: A (advertisement), W (withdrawal), and U (last update).

4.1.5 Flaps Identified by Individual Nodes

The key difference between the three RFD algorithms is the way they identify route flaps. In this subsection, we examine the number of flaps (considering all peers) identified by each BGP node for the three RFD algorithms. We also examine the maximum number of flaps a BGP node identifies from a single peer.

Figure 4.16 illustrates the total number of flaps that each BGP node identifies from all peers in a 100-node network (see Figures B.12 – B.15 in Appendix B for networks of 200 – 500 nodes). It shows that *original RFD* causes each node to report more flaps than *selective RFD* or *RFD+*. *RFD+* reports the least number of flaps. Some nodes report much more flaps than others because they have more connected peers.

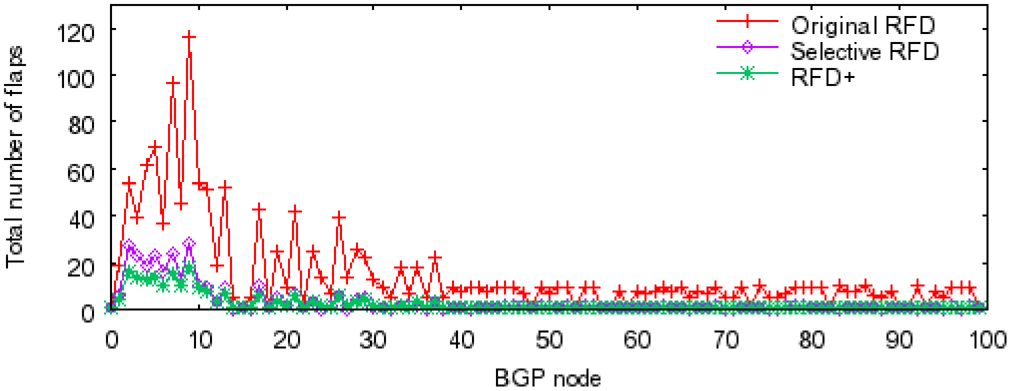

Figure 4.16: Occasional flaps: total number of flaps reported by individual BGP nodes in a 100-node network.

The maximum number of flaps reported by a node for a single peer (i.e., the highest number of flaps caused by a single peer) in various networks is shown in Figure 4.17. The maximum number of flaps that each BGP node reports from one of its peers is shown in Figure 4.18 (a 200-node network) and Figure 4.19 (a 500-node network). Similar figures (Figures B.16, B.18, and B.19) can be found in Appendix B for networks of 100, 300, and 400 nodes.

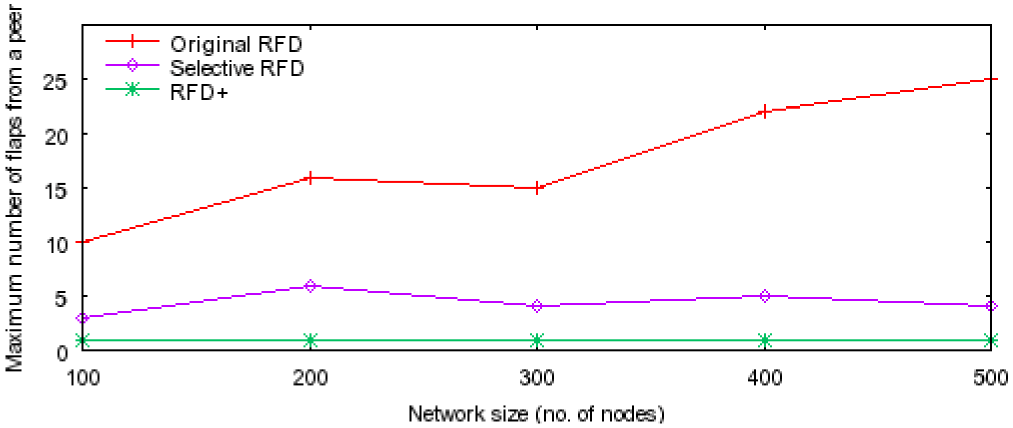

Figure 4.17: Occasional flaps: maximum number of flaps reported by a BGP node for a single peer in various networks.

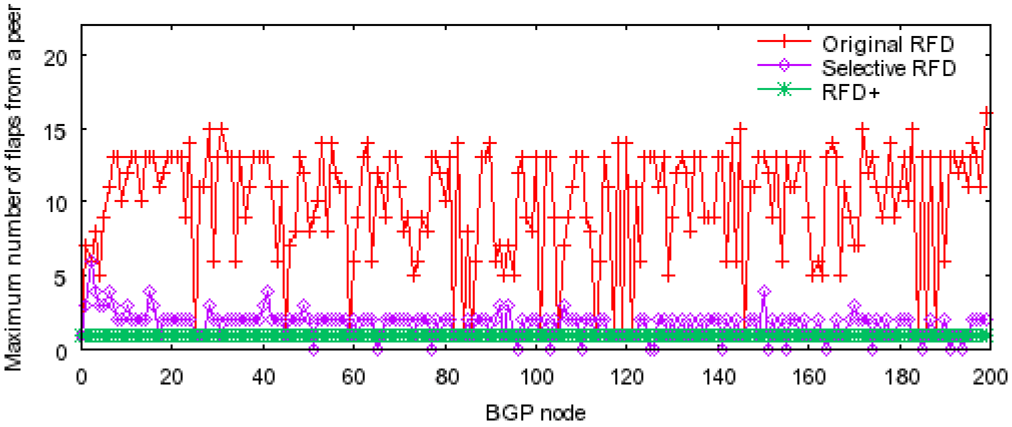

Figure 4.18: Occasional flaps: maximum number of flaps reported by each BGP node for one of its peers in a 200-node network.

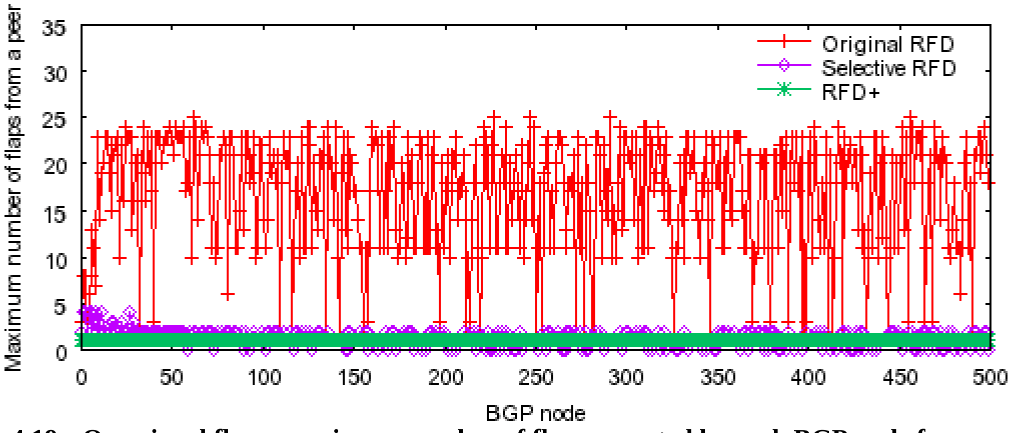

Figure 4.19: Occasional flaps: maximum number of flaps reported by each BGP node for one of its peers in a 500-node network.

***Original RFD*:** Withdrawal-triggered suppression may become severe in *original RFD* because a single flap may cause as many as 25 flaps in a BGP node, as shown in the 500-node network (Figures 4.17 and 4.19). The number of flaps tends to increase as the size of the network increases, as shown in Figure 4.17. In *original RFD*, all nodes recognize and report the flap from their peers (at least 1 flap) when a route withdrawal occurs (Figures 4.18 and 4.19). Figures 4.18 and 4.19 also indicate that the location of a node reporting the largest number of flaps associated with a single peer may be different for different RFD algorithms.

***Selective RFD*:** *Selective RFD* works much better than *original RFD* because a single flap only causes as many as 6 flaps (in the 200-node network, as shown in Figures 4.17 and 4.18) elsewhere. Nevertheless, 6 flaps are sufficient to cause route suppression. In *selective RFD*, some nodes may not report any flaps (0 flap) when a single withdrawal occurs, as shown in Figures 4.18 and 4.19. This is because *selective RFD* postpones counting a withdrawal as a flap until the next update is processed.

***RFD+*:** *RFD+* performs the best because a single flap causes at most one flap in a BGP node, as shown in Figure 4.17. No route suppression occurs in *RFD+* because 1 flap is not sufficient to cause route suppression. *RFD+* does not misinterpret path explorations as route flaps. In *RFD+*, all nodes recognize and report flaps (exactly 1 flap) in the case of a single route withdrawal (Figures 4.18 and 4.19).

4.2 Network Topologies Built from Routing Tables

In addition to BRITE-generated topologies, we also use network topologies built from genuine routing tables that are collected from resources such as RouteViews [36].

We simulate networks with 29 and 110 nodes [33]. They represent connected sub-graphs of the genuine AS graphs of the Internet. Convergence times for individual nodes and the maximum number of flaps associated with a single peer for various RFD algorithms are shown in Figures 4.20 – 4.23. A comparison of convergence time, average time, and numbers of updates, flaps, and suppressions is given in Table 4.3. We use an inter-arrival time of 500 s and 1,000 s for the networks with 29 and 110 nodes, respectively. Figures 4.20 – 4.23 and Table 4.3 show similar results to those obtained for networks generated by BRITE.

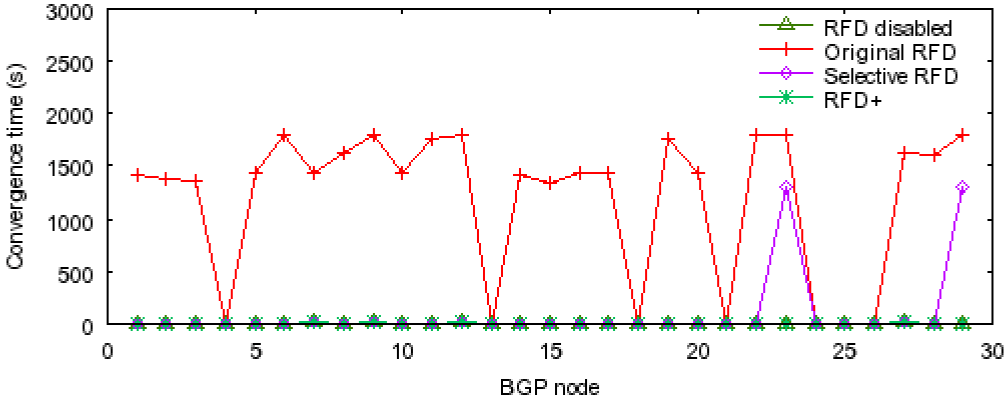

Figure 4.20: Occasional flaps: convergence times for individual BGP nodes in the 29-node network.

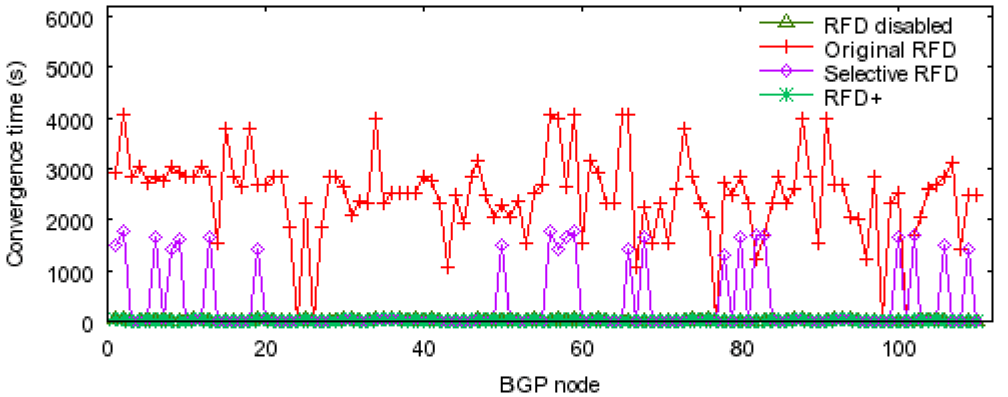

Figure 4.21: Occasional flaps: convergence times for individual BGP nodes in the 110-node network.

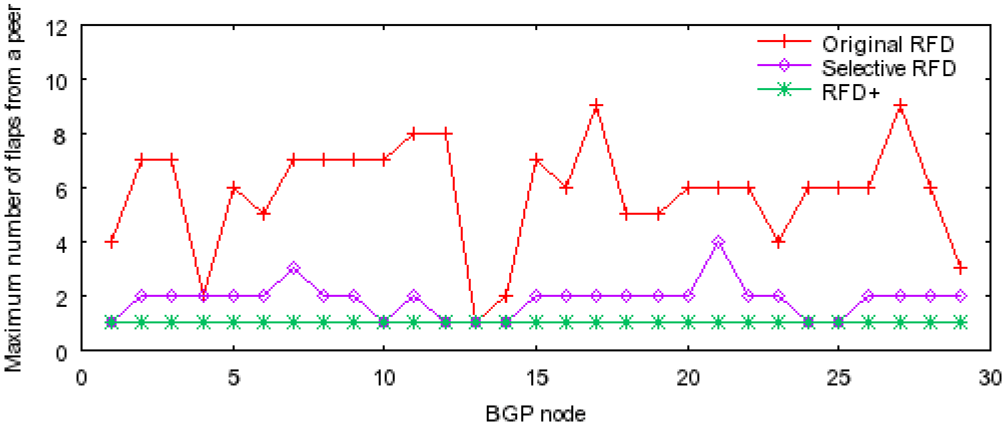

Figure 4.22: Occasional flaps: maximum number of flaps reported by each BGP node for one of its peers in the 29-node network.

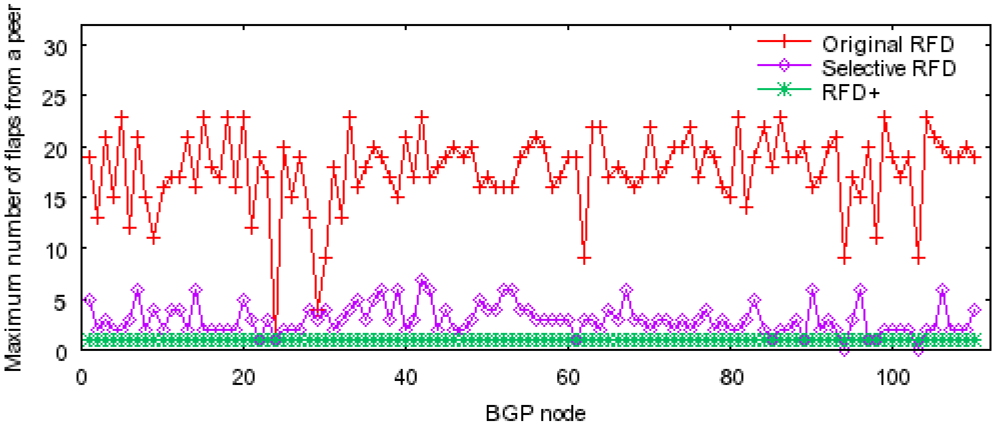

Figure 4.23: Occasional flaps: maximum number of flaps reported by each BGP node for one of its peers in the 110-node network.

Table 4.3: Occasional flaps: comparison between RFD algorithms for networks built from BGP routing tables.

Algorithm	Evaluation parameters	Network size (no. of nodes)	
		29	110
RFD disabled	Convergence time (s)	27.05	27.06
	Average time (s)	3.77	8.39
	No. of updates	1209	17621
	No. of flaps	-	-
	No. of suppressions	-	-
Original RFD	Convergence time (s)	1793.15	4077.29
	Average time (s)	1195.05	2476.27
	No. of updates	1149	11514
	No. of flaps	460	6556
	No. of suppressions	58	489

Algorithm	Evaluation parameters	Network size (no. of nodes)	
		29	110
Selective RFD	Convergence time (s)	1295.44	1761.16
	Average time (s)	93.10	324.82
	No. of updates	1211	17616
	No. of flaps	141	959
	No. of suppressions	1	29
RFD+	Convergence time (s)	27.05	27.06
	Average time (s)	3.77	8.39
	No. of updates	1209	17621
	No. of flaps	91	479
	No. of suppressions	0	0

Original RFD**:** In the case of *original RFD*, one occasional flap causes 58 and 489 route suppressions for networks with 29 and 110 nodes, respectively. As a result, ~75% (~95%) of nodes suffer from a delayed convergence in the 29-node (110-node) network, as shown in Figure 4.20 (Figure 4.21). In the network with 110 nodes, the advertisement of suppressed routes after reuse causes new flaps and suppressions in the network, leading to rather long convergence time. Figure 4.23 shows that a single flap may cause a BGP node to report as many as 23 flaps from a single peer (in the 110-node network). In *original RFD*, network topology affects the number of reported flaps and suppressions. For example, the maximum number of flaps a BGP node reports from a single peer depends on the network size and whether the network topology is dense or sparse. In a densely connected network, a node is connected to many other nodes. In a sparsely connected network, there are many leaf nodes with only one connection. A densely connected network often produces more updates, flaps, and suppressions than a sparsely connected network. It also tends to increase the convergence time during the withdrawal phase. In the 29-node and 110-node networks, which are more densely connected than the networks generated by BRITE, the convergence times for the withdrawal phase are 162.33 s and 648.42 s, respectively.

Selective RFD**:** S*elective RFD* causes much fewer route suppressions than *original RFD* does. There are only 1 and 29 route suppressions in the 29-node and 110-node networks, respectively. However, ~7% (~20%) of the nodes still suffer from a delayed convergence (over 20 min) in the 29-node (110-node) network, as shown in Figure 4.20 (Figure 4.21). Figure 4.23 shows that the maximum number of flaps reported by a node for a single peer is reduced to 7 (in the 110-node network) compared to *original RFD*. Nevertheless, there are still ~25% of the nodes that report at least 4 flaps from one of the peers, as indicated by points corresponding to a value ≥ 4 in Figure 4.23. Route suppression may occur with 4 flaps. As is the case with *original RFD*, network topology may affect the number of flaps and suppressions in *selective RFD*.

In general, damping helps to reduce the number of generated update messages. However, this is not always the case. For example, in the 29-node network, *selective RFD* produces 2 additional update messages compared to the case when RFD is disabled, as shown in Table 4.3. This is due to the additional updates caused by the reuse and advertisement of suppressed routes in *selective RFD*.

RFD+**:** *RFD+* has superior performance in the case of a single flap because the occasional flap does not lead to any route suppression in either the 29-node network or the 110-node network, as shown in Table 4.3. Figures 4.22 and 4.23 show that the maximum number of flaps reported by a BGP node for a single peer is always 1, regardless of the network size. *RFD+* can correctly identify the occasional flap and does not misinterpret path explorations as flaps.

4.3 Summary

Based on the presented simulation results in relatively large and realistic network topologies, the key observations regarding BGP behavior and the performance of various RFD algorithms in the case of occasional flaps are:

- A withdrawal message takes significantly longer to converge than an advertisement message. This holds for all damping algorithms. However, *original RFD* has the fastest convergence during the withdrawal phase because of its most aggressive route suppression.
- The convergence time of the withdrawal phase depends heavily on network size and network topology (dense or sparse). However, the convergence time of the advertisement phase does not depend much on network size and may not vary with the increase of network size.
- There is no visible relationship, such as monotonic increase or decrease, between inter-arrival time and BGP convergence time. Beyond a certain threshold, further increase in inter-arrival time does not affect convergence time. The time difference between the instances when an update is ready to be sent and when the MRAI timer expires may strongly affect the length of convergence time. This is evident especially when the inter-arrival time between updates is short.

- RFD algorithms have little effect on convergence time and number of updates when inter-arrival time is rather short. This is because there are no or only few route suppressions in the network.

- The effect of origin router's location on convergence time and number of updates depends on network topology, the phase (advertisement or withdrawal), and the RFD algorithm. Nevertheless, when the origin router is located at the edge of the network, the convergence time and number of updates often increase compared to the case when the origin router is located at the core of the network.

- *Original RFD* does not perform well in the case of occasional flaps. A single flap may significantly delay the convergence due to *original RFD*'s aggressive route suppression.

- *Selective RFD* and *RFD+* perform better than *original RFD* in identifying genuine route flaps. *RFD+* has the best performance and reports no additional flaps in all simulated cases.

5 PERFORMANCE ANALYSIS OF RFD ALGORITHMS: PERSISTENT FLAPS

The RFD mechanism is targeted at persistent flaps. We use a series of 5 flaps with an inter-arrival time of 300 s to simulate persistent flaps. Considered network topologies are generated by BRITE or built from BGP routing tables. As in Section 4， we examine the following variables:

- total number of update messages generated by all BGP speakers
- total number of flaps reported for the entire network
- maximum number of flaps associated with a single peer of each BGP speaker
- total number of route suppressions caused by all flaps in the network
- convergence time.

The simulation results are given in Subsection 5.1. Subsection 5.2 is a summary of the key observations.

5.1 Performance of RFD Algorithms

Convergence times of individual BGP nodes for various RFD algorithms are shown in Figure 5.1 (100-node network) and Figure 5.2 (200-node network) for BRITE-generated topologies, and in Figure 5.3 (29-node network) and Figure 5.4 (110-node network) for topologies built from routing tables. The convergence time, average time, and numbers of updates, flaps, and suppressions for various RFD algorithms are shown in

Table 5.1 (for BRITE-generated topologies) and Table 5.2 (for topologies built from routing tables). The maximum number of flaps associated with a single peer is shown in Figure 5.5 (300-node network) and Figure 5.6 (110-node network). Appendix C contains relevant figures and tables that are supplementary to Figures 5.1 – 5.6 for all simulated network topologies in the case of persistent flaps. These supplementary figures and tables show results that agree with those obtained from Figures 5.1 – 5.6.

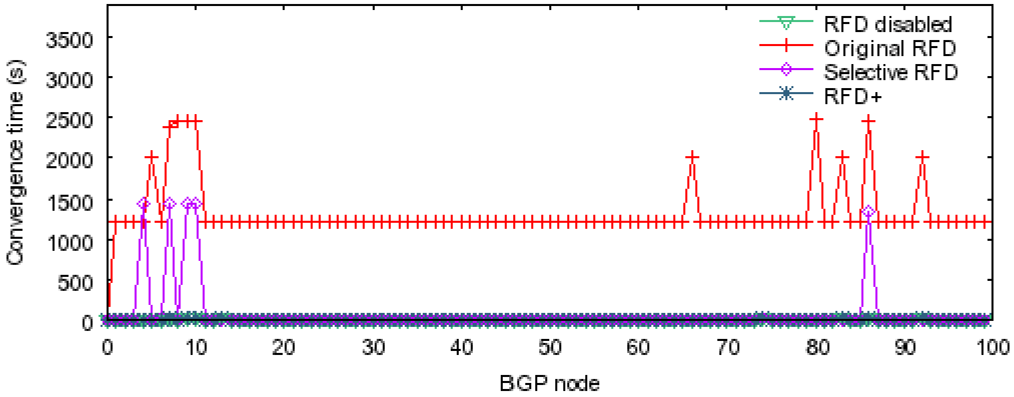

Figure 5.1: Persistent flaps in BRITE-generated topologies: convergence times for individual BGP nodes in a 100-node network.

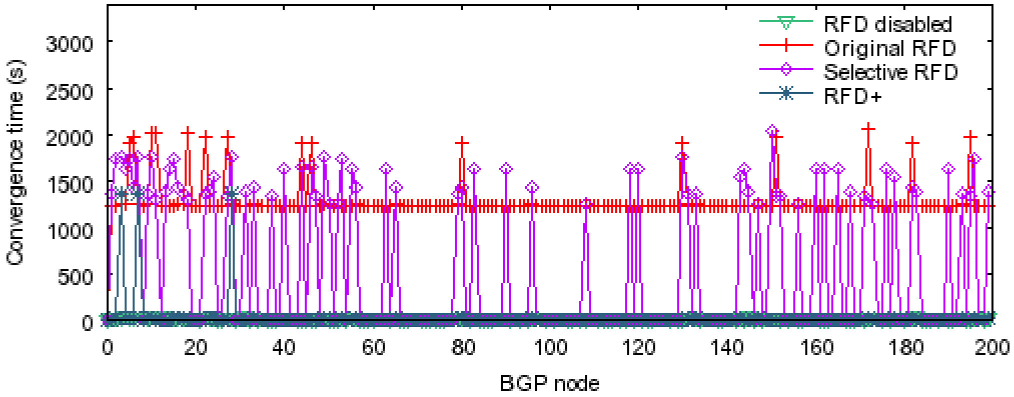

Figure 5.2: Persistent flaps in BRITE-generated topologies: convergence times for individual BGP nodes in a 200-node network.

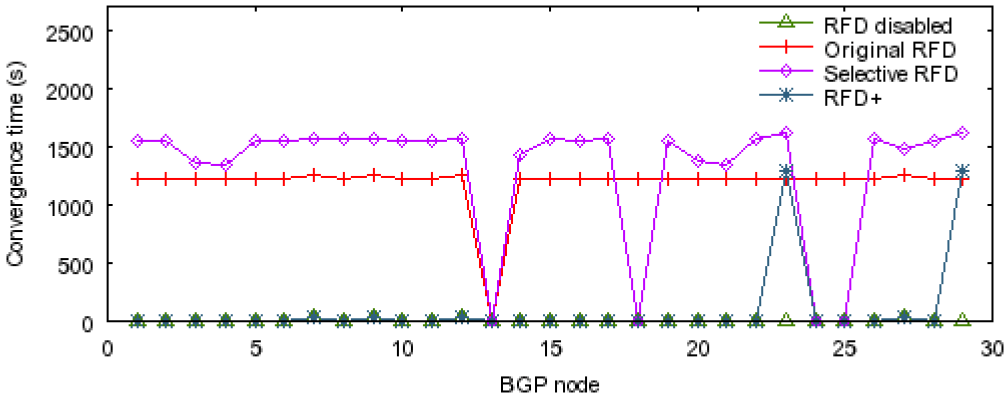

Figure 5.3: Persistent flaps in topologies built from routing tables: convergence times for individual BGP nodes in a 29-node network.

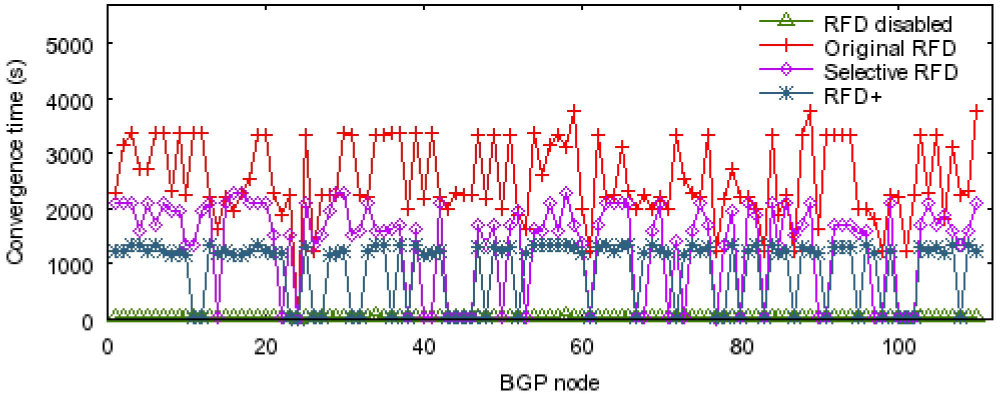

Figure 5.4: Persistent flaps in topologies built from routing tables: convergence times for individual BGP nodes in a 110-node network.

Table 5.1: Persistent flaps in BRITE-generated topologies: comparison between various RFD algorithms.

Algorithm	Evaluation parameters	Network size (no. of nodes)				
		100	200	300	400	500
RFD disabled	Convergence time (s)	27.02	51.02	51.02	51.02	51.02
	Average time (s)	2.175	14.34	14.37	18.23	19.21
	No. of updates	15935	38052	53789	75674	104236
	No. of flaps	-	-	-	-	-
	No. of suppressions	-	-	-	-	-
Original RFD	Convergence time (s)	2487.3	2048.8	3023.2	3545.55	4213.68
	Average time (s)	1320.47	1279.32	1490.44	1504.94	1588.30
	No. of updates	8016	11784	30822	53271	78519
	No. of flaps	4480	6561	19961	35288	52888
	No. of suppressions	256	487	802	1038	1385

Algorithm	Evaluation parameters	Network size (no. of nodes)				
		100	200	300	400	500
Selective RFD	Convergence time (s)	1437.03	2035.75	1734.5	1797.43	1977.6
	Average time (s)	71.53	531.22	205.45	222.76	246.32
	No. of updates	13251	32535	52852	70924	100726
	No. of flaps	2514	5209	7720	9867	13184
	No. of suppressions	172	411	623	763	1027
RFD+	Convergence time (s)	27.02	1369.21	27.02	51.02	51.01
	Average time (s)	2.175	24.60	14.27	19.19	17.68
	No. of updates	15850	37569	53452	75205	103892
	No. of flaps	856	1642	2557	3391	4417
	No. of suppressions	44	91	100	121	201

Table 5.2: Persistent flaps in topologies built from routing tables: comparison between various RFD algorithms.

Algorithm	Evaluation parameters	Network size (no. of nodes)	
		29	110
RFD disabled	Convergence time (s)	27.05	78.05
	Average time (s)	3.77	43.78
	No. of updates	5109	45607
	No. of flaps	-	-
	No. of suppressions	-	-
Original RFD	Convergence time (s)	1255.74	3774.23
	Average time (s)	1190.09	2535.86
	No. of updates	1675	18308
	No. of flaps	716	11467
	No. of suppressions	94	538
Selective RFD	Convergence time (s)	1616.79	2295.76
	Average time (s)	1311.21	1385.71
	No. of updates	3291	44568
	No. of flaps	843	4989
	No. of suppressions	63	385
RFD+	Convergence time (s)	1292.47	1335.67
	Average time (s)	92.90	919.61
	No. of updates	5106	45694
	No. of flaps	342	1863
	No. of suppressions	13	128

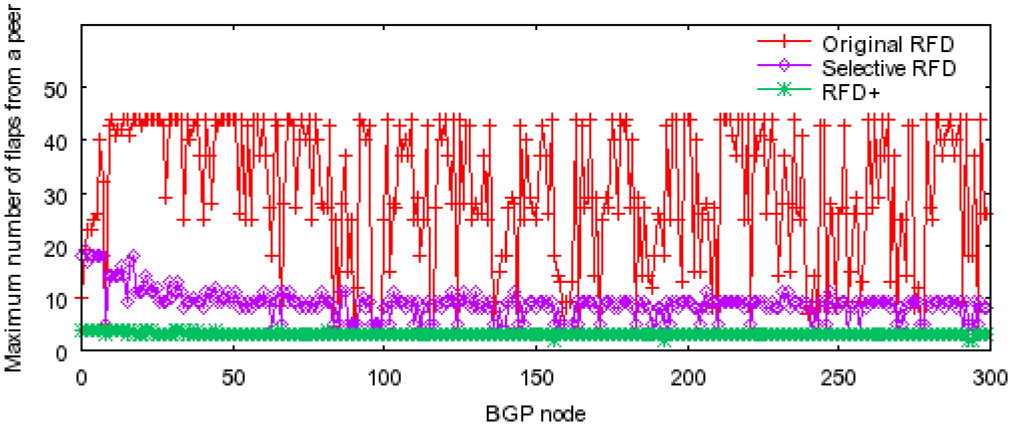

Figure 5.5: **Persistent flaps in BRITE-generated topologies: maximum number of flaps reported by each BGP node for one of its peers in a 300-node network.**

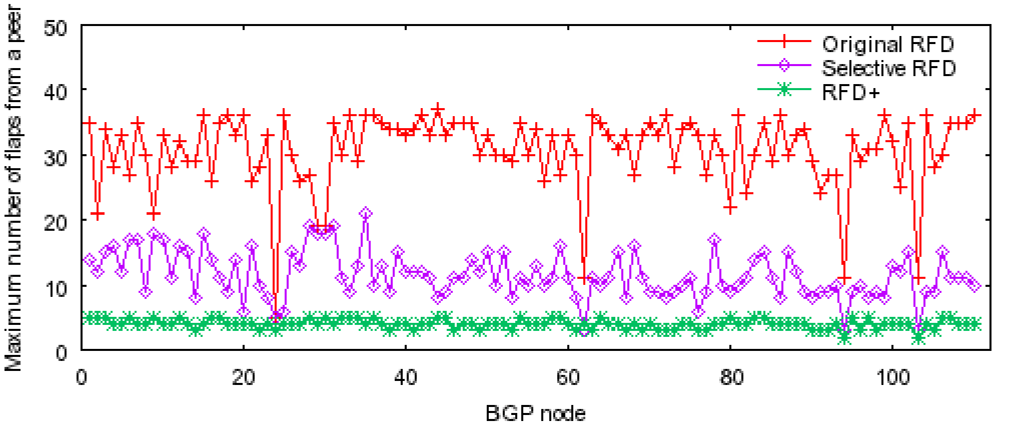

Figure 5.6: **Persistent flaps in topologies built from routing tables: maximum number of flaps reported by each BGP node for one of its peers in a 110-node network.**

RFD disabled**:** *RFD disabled* has the shortest convergence time because no suppression period is involved, as shown in Figures 5.1 – 5.4. However, the shortest convergence time is achieved at the cost of the largest number of update messages, as shown in Tables 5.1 and 5.2.

Original RFD**:** *Original RFD* results in the longest convergence time because extensive route suppression occurs and the suppression period is long, as shown in Figures 5.1 – 5.4. This is expected because route flap damping is designed to suppress routes that flap persistently. As is the case of occasional flaps, *original RFD* is the most aggressive in suppressing persistently flapping routes. Five flaps in *original RFD* are sufficient to suppress the flapping route in all simulated cases, as shown in Tables 5.1 and 5.2. All the nodes except the one directly connected to the origin router suffer from a significantly delayed convergence (convergence time > 1,000 s) due to the route suppression, as shown in Figures 5.1 – 5.4. However, this is not the case with *selective RFD* and *RFD+*.

Original RFD performs the best in achieving network stability in the case of persistent flaps. It suppresses persistent route flaps as early as possible. As a result, it always produces the least number of update messages in the network. This is shown in Figure 5.7, which illustrates the total number of updates in various networks for various RFD algorithms. On some occasions, the decrease in the number of updates is quite significant. For example, in the 200-node network, *original RFD* generates ~70% less updates than *RFD+*.

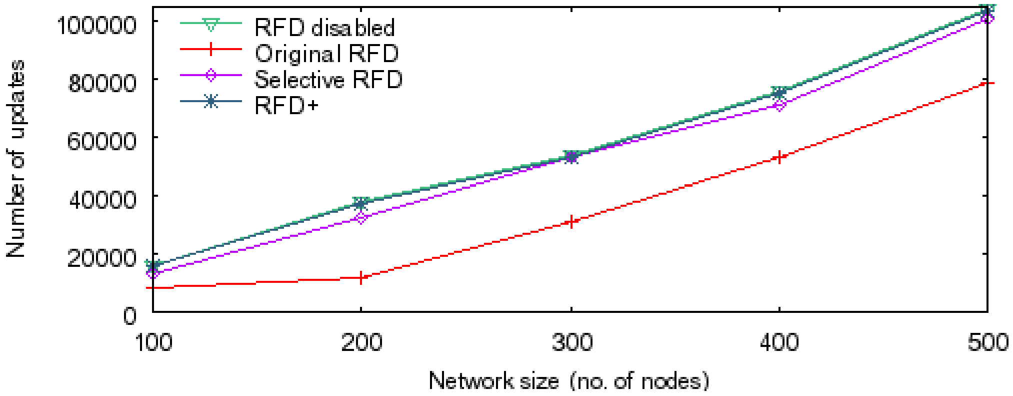

Figure 5.7: Persistent flaps in BRITE-generated topologies: total number of updates in various networks.

***Selective RFD*: **In the topologies generated by BRITE, over 65% of the nodes do not suffer from a delayed convergence caused by route suppression (Figures 5.1 and 5.2), even though the route flaps 5 times within a relatively short time. In some cases, the percentage of such nodes even reaches ~95%, as shown in Figure 5.1. For topologies built from routing tables where nodes are more densely connected, a higher percentage of nodes suffer from a delayed convergence due to the persistent flaps. Nevertheless, ~15% and ~25% of the nodes do not exhibit the effect of route suppression (caused by persistent flaps) in the 29-node (Figure 5.3) and 110-node (Figure 5.4) networks, respectively. In all cases, the node directly connected to the origin router does not suppress the flapping route after reporting 5 flaps. This is because *selective RFD* postpones counting a withdrawal as a flap. When a node receives a withdrawal, it only remembers the withdrawal penalty temporarily, without considering suppression. If the next update is identified as a flap, it will restore the overall penalty by adding the temporary withdrawal penalty. Since the temporary withdrawal penalty decays exponentially over time, its addition to the overall penalty at a later time may not be sufficient to suppress the flapping route. Therefore, *selective RFD* may require the occurrence of additional flaps in order to suppress a flapping route. For example, when the inter-arrival time between updates is 300 s, *original RFD* requires the occurrence of 4 flaps in order to suppress the flapping route, while *selective RFD* will wait for 6 flaps before suppressing the route. This situation worsens when the inter-arrival time between updates increases. Because of this property of *selective RFD*, it is possible for a node to report more flaps, accumulate higher penalty, and cause longer suppression in *selective RFD* than in *original RFD*, as shown in Figure 5.3. *Selective RFD* may cause a delay in route suppression.

***RFD+*:** *RFD+* is the least aggressive in suppressing persistently flapping routes. In networks generated by BRITE, no route suppression occurs and no delayed convergence is experienced except for one case where only ~2% of the nodes suffer from the suppression-triggered delay, as shown in Figure 5.2. In networks built from routing tables, ~7% and ~70% of the nodes experience a delayed convergence due to route suppression in the 29-node (Figure 5.3) and 110-node (Figure 5.4) networks, respectively. In the 110-node network, *RFD+* generates more updates than the case when RFD is disabled, as shown in Table 5.2. This is due to the additional updates caused by the reuse and re-advertisement of suppressed routes in *RFD+*. An important observation about *RFD+* is that the peer of the origin router and many other nodes (~90% in Figure 5.5 and ~20% in Figure 5.6) report only 3 flaps instead of 5 flaps. Three flaps are not sufficient to suppress a flapping route according to the RFD implementation. *RFD+* considers a series of 5 updates (*advertisement, withdrawal, re-advertisement, withdrawal, and re-advertisement again*) as only 1 flap rather than 2 flaps. The last 2 updates are not sufficient to cause an additional flap. Even though *RFD+* has the advantage of reporting no additional flaps under the circumstances of both occasional and persistent flaps, it may underestimate the number of genuine route flaps, causing a delay in route suppression.

5.2 Summary

The summary of the key observations regarding the performance of RFD algorithms in the case of persistent flaps is:

- *Original RFD* performs the best in preventing the spread of routing oscillations as early as possible. *Selective RFD* may require the occurrence of additional flaps in order to suppress a flapping route, and this situation worsens as the inter-arrival time between updates increases. *RFD+* underestimates the number of genuine route flaps and delays route suppression. *Selective RFD* and *RFD+* are more lenient than *original RFD* in suppressing persistently flapping routes. This is not a desirable property because it may cause a higher number of updates to be generated and a heavier processing load to be imposed on BGP speakers.

- *Original RFD* is optimal in reducing the number of updates because it always leads to the least number of updates, even though it may significantly delay the BGP convergence. This decrease in the number of updates due to *original RFD* is rather significant in some cases.

6 PROPOSED IMPROVEMENTS OF RFD ALGORITHMS

After identifying problems regarding the current RFD algorithms, we propose two possible improvements.

6.1 Weaknesses of the Existing RFD Algorithms

According to RFC 2439 [14], a well-designed route flap damping algorithm should act efficiently against persistent route flaps and at the same time be “tolerant” to occasional route flaps. Simulation results presented in Sections 4 and 5, based on relatively large and realistic network topologies, show that o*riginal RFD* behaves more favorably in the case of persistent flaps, while *selective RFD* and *RFD+* perform better in the case of occasional flaps. *Original RFD* behaves aggressively against route flaps, which leads to the early suppression of persistent flaps and efficient reduction of update messages. *Selective RFD* and *RFD+* can better identify genuine route flaps and do not tend to suppress a route that flaps occasionally.

As described in Subsection 5.1, the way *selective RFD* deals with withdrawal messages poses a problem: *selective RFD* may require the occurrence of additional flaps in order to suppress a persistently flapping route, causing a delay in route suppression. We calculated the number of flaps required to suppress a flapping route for both *original* and *selective RFD* algorithms, based on the default Cisco settings for RFD parameters as shown in Table 2.1. The calculation process is:

1. The initial value of penalty is set to 0.
2. When a route flap is identified, it incurs a penalty of 1,000 (for withdrawals) or 500 (for route attribute changes). The number of flaps is incremented by 1.
3. The penalty decays exponentially over time according to:

   $$penalty(t2) = penalty(t1) * e^{(-(t2 - t1) * ln2 / half\text{-}life)},$$

   where penalty(t1) and penalty(t2) are the penalty values at times t1 and t2, respectively (t2 > t1). Half-life is set to 900 s.
4. When penalty is over 2,000, the route becomes suppressed.

Note that *selective RFD* postpones counting a withdrawal as a flap and only remembers the temporary withdrawal penalty. The temporary withdrawal penalty decays exponentially over time, in the same way as the overall penalty of a route does. If the next update is identified as a flap, the decayed withdrawal penalty will be added to the overall penalty of the route.

The relationship between inter-arrival time and number of flaps required to suppress a route is shown in Table 6.1. As the inter-arrival time between updates increases, *selective RFD* waits for a significant number of flaps (as high as 14 flaps, as shown in Table 6.1) before suppressing a flapping route. The calculation of the required number of flaps to suppress a route also suggests that *selective RFD* will not suppress any route if the inter-arrival time between updates is larger than 322 s.

Table 6.1: Comparison between *selective RFD* and *original RFD* regarding inter-arrival time and number of flaps required to suppress a route.

Inter-arrival time between updates (s)	*Selective RFD*	*Original RFD*
100	3	3
150	3	3
200	4	3
250	4	3
260	5	3
270	5	3
280	5	3
290	6	3
300	6	3
305	7	3
310	7	3
312	8	3
314	8	4
316	9	4
318	10	4
320	12	4
321	14	4

RFD+ suffers from another problem: it underestimates the number of genuine route flaps. This delays the suppression of persistently flapping routes. As described in Subsection 5.1, *RFD+* considers a series of 5 updates (*advertisement, withdrawal, re-advertisement, withdrawal, and re-advertisement*) as only 1 flap, rather than 2 flaps. A peer of the origin router will report only *floor((N+1)/2)* number of flaps if the origin router experiences a failure (down) followed by a recovery (up) for *N* consecutive times. Another pitfall of *RFD+* is its potentially large memory consumption because a BGP speaker needs to store all the interim routes during path exploration for each prefix from each peer. The number of prefixes in the Internet is ~130,000 [37] and a router at the core of the Internet may have ~100 peers. A conservative guess is that if each route takes ~20 bytes of memory and the number of interim routes for each prefix is ~15, then ~4 gigabytes (130,000 * 100 * 20 * 15 bytes) of memory would be required. This is a non-trivial amount for a router.

6.2 *Modified RFD+* Algorithm

We propose a simple modification to *RFD+* by keeping track of the existence of the "up-down-up" state of a route. The tracking of a route's "up-down-up" state is also used by *selective RFD*. A flap is identified either when detected by the original *RFD+* algorithm or when a route is advertised, withdrawn, and advertised again. The pseudo code of the *modified RFD+* algorithm is shown in Algorithm 6.1. With this simple modification, a peer of the origin router could identify all *N* flaps when the origin router fails and then recovers for *N* consecutive times. A series of 5 updates (*advertisement, withdrawal, re-advertisement, withdrawal, and re-advertisement*) are now considered as 2 flaps. The comparison between the *RFD+* algorithm and its modified version in terms of convergence time and numbers of updates, flaps, and suppressions is shown in Table 6.2. The comparison is illustrated in Figure 6.1 (convergence time) and Figure 6.2 (number of updates). The inter-arrival time between updates is 300 s. *Modified RFD+* has a much longer convergence time because the flapping route is suppressed by the peer of the origin router in all cases. This is not the case for *RFD+*, where nodes do not suffer from suppression of routes (except in the 200-node network). As a result of the route suppression, the number of updates is reduced (by up to ~20%) in the case of the *modified RFD+* algorithm. The suppression of a persistently flapping route is the desired behavior. This is what the route flap damping mechanism is for. Simulation results also suggest that in rare cases the *modified RFD+* algorithm may cause a BGP node to report additional flaps. Nevertheless, *modified RFD+* behaves better than other RFD algorithms in identifying genuine route flaps when both occasional and persistent flaps are considered. *Modified RFD+* should be used when the main concern is to identify route flaps properly, even though it is not as aggressive as *original RFD* in reducing the number of update messages.

```
when receiving a route r with prefix d from peer j
if (W(r))                          // W(x) returns true only if x is a withdrawn route
        preUpdate = 0              // just remember the update type without doing anything
                                   // else; 0 and 1 indicate withdrawal and advertisement
                                   // respectively
else                               // current route r is an advertisement
        if (preUpdate == 0 and dop(r) == preDop)        // ‘up-down-up’ state is detected
                                   // dop(x) returns the degree of preference of route x
              a flap is identified
              clear R(d, j)         // R(d, j) is the set of all routes with prefix d announced
        else                        // from peer j
              if (r ∉ R(d, j))
                    insert r to the set R(d, j)
              else                  // r is in the set R(d, j)
                    if (preUpdate == 0)
                          a flap is identified
                          clear R(d, j)
                    else
                          if (dop(r) > preDop)
                                a flap is identified
                                clear R(d, j)
        preDop = dop(r)             // remember both the degree of preference and update
        preUpdate = 1               // type of route r
```

Algorithm 6.1: Pseudo code of the *modified RFD+* algorithm.

Table 6.2: Persistent flaps in BRITE-generated topologies: comparison between the original *RFD+* and the *modified RFD+* algorithms in the case of 5 flaps.

Algorithm	Evaluation parameters	Network size (no. of nodes)				
		100	200	300	400	500
Modified RFD+	Convergence time (s)	1555.71	1555.71	1555.71	1555.71	1555.71
	No. of updates	12805	30541	45859	68038	98194
	No. of flaps	692	1335	2016	2684	3417
	No. of suppressions	5	22	31	22	23
RFD+	Convergence time (s)	27.02	1369.21	27.02	51.02	51.01
	No. of updates	15850	37569	53452	75205	103892
	No. of flaps	856	1642	2557	3391	4417
	No. of suppressions	44	91	100	121	201

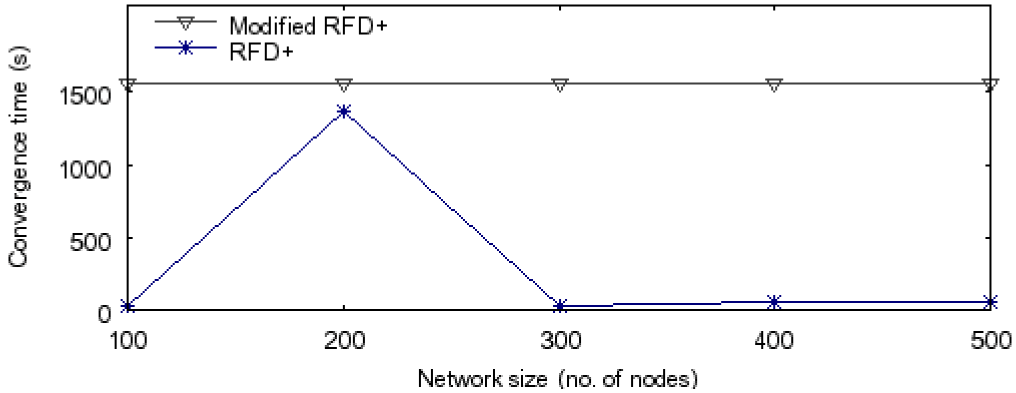

Figure 6.1: Persistent flaps in BRITE-generated topologies: comparison of convergence time between *modified RFD+* and *RFD+*.

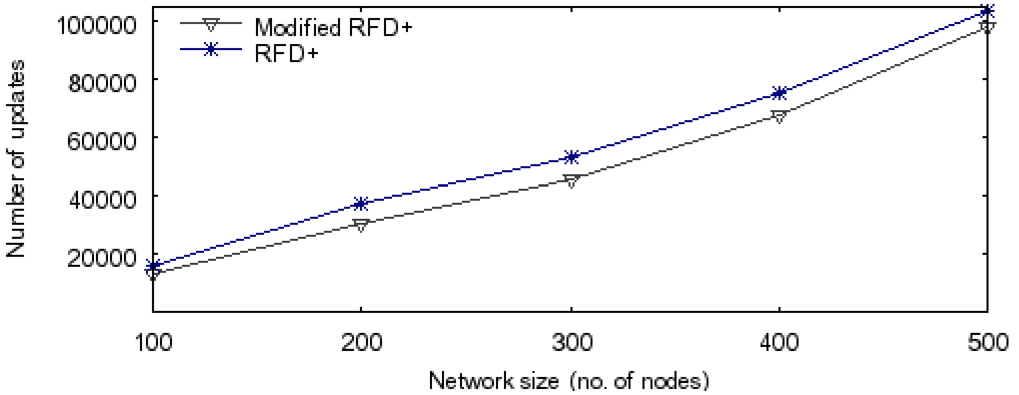

Figure 6.2: Persistent flaps in BRITE-generated topologies: comparison of number of updates between *modified RFD+* and *RFD+*.

One way to reduce the memory consumption for *RFD+* is to hash each interim route into a simpler data type (e.g., integer) and store it rather than storing the complete route. If storing the complete information of a route requires 20 bytes, then hashing may reduce a router's memory consumption by 80% because storing an integer requires only 4 bytes. Thus, the potential memory consumption of a router at the core of the Internet would be substantially reduced from ~4 gigabytes to ~800 megabytes. Since the space of integers is much larger than the number of different interim routes we need to store, proper hashing is achievable. Hashing may also reduce a router's processing time since comparing two integers is faster than comparing two routes. It is effective because *RFD+* requires a large number of comparisons between routes.

6.3 An Adaptive RFD Approach

An effective route flap damping algorithm is important in maintaining the stability of the Internet as various kinds of network instabilities exist. Ski Ilnicki and Alexander Tudor [38] recorded as high as ~600 flaps during a particular hour on the day Aug.31, 2002 on some of their routers, and the ratio of total flaps over total updates could reach as high as ~30% for a certain day during the period from Aug.1, 2001 to Jan.31, 2002. The following tables that we built, based on the genuine routing information from Route Views data archives [39]-[41], list all the update messages between two randomly-picked BGP speakers regarding a certain prefix (also randomly picked) within a 15-minute period. For simplicity, we only list the AS-path attribute of the route in the tables.

Table 6.3: A list of all updates between two BGP nodes for prefix 202.20.105.0/24 within a 15-minute period. A (Advertise) and W (Withdraw).

Prefix: 202.20.105.0/24 From: 67.17.80.219 – AS3549	Time Period: 21:59 09/01/2004 – 22:14 09/01/2004 To: 128.223.60.102 – AS6447	
Time	AS-path	Update
21:59:01	3549 4637 7693	A
21:59:13	3549 3561 3491 9304 4651 4651 4651 4651 4651 7693	A
21:59:13	3549 3561 3561 3786 3786 4651 4651 4651 7693	A
21:59:13	3549 3561 3491 9304 4651 4651 4651 4651 4651 7693	A
21:59:35		W
21:59:35	3549 3561 3491 9304 4651 4651 4651 4651 4651 7693	A
21:59:35		W
21:59:35	3549 3561 3561 3491 9304 4651 4651 4651 4651 4651 7693	A
21:59:39		W
22:06:46	3549 4651 7693	A
22:10:13	3549 6453 4651 7693	A
22:10:13	3549 10026 9304 4651 4651 4651 4651 4651 7693	A
22:10:13	3549 6453 4651 7693	A
22:10:13	3549 10026 9304 4651 4651 4651 4651 4651 7693	A
22:10:33		W
22:10:34	3549 4637 7693	A
22:10:57	3549 4651 7693	A

Table 6.4: A list of all updates between two BGP nodes for prefix 128.109.0.0/16 within a 15-minute period. A (Advertise) and W (Withdraw).

Prefix: 128.109.0.0/16 From: 67.17.80.219 – AS3549	Time Period: 00:17 09/02/2004 – 00:32 09/02/2004 To: 128.223.60.102 – AS6447	
Time	AS-path	Update
00:17:34	3549 209 81	A
00:17:34		W
00:17:34	3549 2914 3948 11537 81	A
00:17:34		W
00:17:34	3549 2914 3948 24 11537 81	A
00:17:34	3549 209 81	A
00:17:34	3549 2914 3948 11537 81	A
00:17:50	3549 209 81	A
00:17:56		W
00:17:56	3549 2914 3948 11537 81	A
00:18:07		W
00:18:47	3549 209 81	A
00:19:15	3549 2914 3948 11537 81	A
00:19:16	3549 209 81	A
00:19:41		W
00:19:41	3549 2914 3948 11537 81	A
00:19:41	3549 2914 3948 11537 81	A
00:19:41		W
00:20:08	3549 2914 3948 11537 81	A

Prefix: 128.109.0.0/16 From: 67.17.80.219 – AS3549	Time Period: 00:17 09/02/2004 – 00:32 09/02/2004 To: 128.223.60.102 – AS6447	
Time	AS-path	Update
00:20:33	3549 209 81	A
00:21:26	3549 209 81	A
00:21:57	3549 209 81	A
00:24:22	3549 209 81	A
00:25:38	3549 209 81	A
00:26:33		W
00:26:33	3549 2914 3948 11537 81	A
00:26:36		W
00:26:37	3549 209 81	A
00:26:59	3549 2914 3948 11537 81	A
00:27:05	3549 209 81	A
00:27:25		W
00:27:25	3549 2914 3948 11537 81	A
00:27:33		W
00:27:33	3549 209 81	A
00:27:51	3549 2914 3948 11537 81	A
00:29:20	3549 209 81	A
00:30:46	3549 209 81	A
00:32:20		W
00:32:20	3549 2914 3948 11537 81	A
00:32:23		W

Tables 6.3 and 6.4 reflect the level of network instability in the Internet. Persistent flaps are common in the current Internet. The origin of the instability is unknown. If the length of AS-path is used as a metric for route preference, the numbers of reported flaps on the receiving nodes with *original RFD*, *selective RFD*, and *RFD+* are 16, 9, 2 for Table 6.3 and 33, 21, 7 for Table 6.4, respectively. The results given by these damping algorithms vary considerably for the two 15-minute-long events. The adoption of a well-behaved damping algorithm can make a difference on network performance.

While *original RFD* is designed to work well with persistent flaps, *selective RFD* and *RFD+* perform better in the case of occasional flaps. None of these algorithms behaves optimally under all circumstances. There is a trade-off between network stability and availability of routes. Good availability of a route implies no route suppression and, hence, either no damping or a rather lenient damping algorithm should be used. This may cause more update messages to be generated due to lack of suppression of flapping routes and, hence, may result in network instability. In terms of the convergence time, stability demands more aggressive damping algorithms for the reduction of generated update messages. This may result in route suppression and delayed convergence. To achieve a balance between stability and availability, an adaptive approach in route flap damping is desired.

We propose a second modification to the current RFD algorithms. It is a simple adaptive RFD algorithm named *combined RFD*, which integrates the *original RFD* and the *modified RFD+* algorithms. Within a certain period of time, a BGP speaker uses the *modified RFD+* algorithm for the first two identified flaps. It then switches to the *original RFD* algorithm starting with the third flap. The motivation is to take advantage of the strengths of both *original RFD* and *modified RFD+* so that a route will not be suppressed if it flaps only once or twice but will be efficiently suppressed if it flaps persistently. Table 6.5 shows the simulation results when an existing route becomes unavailable and then available repeatedly for 8 consecutive times. The same results are also illustrated in Figure 6.3 (convergence time) and Figure 6.4 (number of updates). The inter-arrival time between updates is set to 120 s. Figure 6.4 shows that *combined RFD* performs the second best in reducing update messages in the case of persistent flaps and it is close to *original RFD*. The number of update messages in the case of *combined RFD* is reduced by up to ~16%, compared to the *modified RFD+* algorithm. The *combined RFD* algorithm also tends to generate fewer updates than *selective RFD* and *RFD+*. In most cases, *combined RFD* generates less than 7% of additional update messages compared to *original RFD*. However, unlike *original RFD*, it does not suppress a route that flaps only occasionally. Although *combined RFD* may not always accurately identify genuine route flaps, it is a proper choice when the main concern is to keep the number of update messages close to optimal (generating as few updates as *original RFD* does). Through a compromise between *original RFD* and *RFD+*, *combined RFD* makes one step forward in achieving a balance between network stability and availability of routes.

Table 6.5: BRITE-generated topologies: comparison between various RFD algorithms when a route flaps 8 times. Inter-arrival time between updates is 120 s.

Algorithm	Evaluation parameters	Network size (no. of nodes)				
		100	200	300	400	500
RFD disabled	Convergence time (s)	42.02	42.02	15.21	42.1	42.02
	Average time (s)	11.78	15.92	13.37	15.49	14.32
	No. of updates	15675	30904	45823	59441	78906
	No. of flaps	/	/	/	/	/
	No. of suppressions	/	/	/	/	/
Original RFD	Convergence time (s)	2908.06	3067.35	3426.2	3705.34	3910.4
	Average time (s)	2240.18	2294.81	2289.67	2295.26	2344.44
	No. of updates	7519	16062	28054	37699	53436
	No. of flaps	4553	10235	19103	25563	36896
	No. of suppressions	261	503	815	1048	1389
Selective RFD	Convergence time (s)	2254.92	2254.92	2254.92	2254.92	2254.92
	Average time (s)	2207.80	2221.23	2226.42	2226.80	2228.91
	No. of updates	8468	16207	33859	40549	60068
	No. of flaps	1874	3803	5952	7610	10479
	No. of suppressions	178	390	663	857	1161
RFD+	Convergence time (s)	1349.38	1349.38	1349.38	1349.38	1502.64
	Average time (s)	1311.31	1320.22	1324.51	1324.11	1326.31
	No. of updates	14344	28673	50167	66713	87916
	No. of flaps	1100	2154	3329	4326	5729
	No. of suppressions	66	120	148	152	238

Algorithm	Evaluation parameters	Network size (no. of nodes)				
		100	200	300	400	500
Modified RFD+	Convergence time (s)	2374.92	2374.92	2374.92	2374.92	2374.92
	Average time (s)	2326.60	2340.63	2346.02	2346.50	2348.67
	No. of updates	9190	19347	30958	45119	65630
	No. of flaps	676	1270	1975	2552	3300
	No. of suppressions	5	17	15	16	15
Combined RFD	Convergence time (s)	2271.41	2271.41	2271.41	2971.71	3583.96
	Average time (s)	2224.13	2237.64	2242.51	2254.83	2289.71
	No. of updates	8202	17204	28608	37716	55477
	No. of flaps	2006	4622	9012	12137	19476
	No. of suppressions	254	499	788	1043	1339

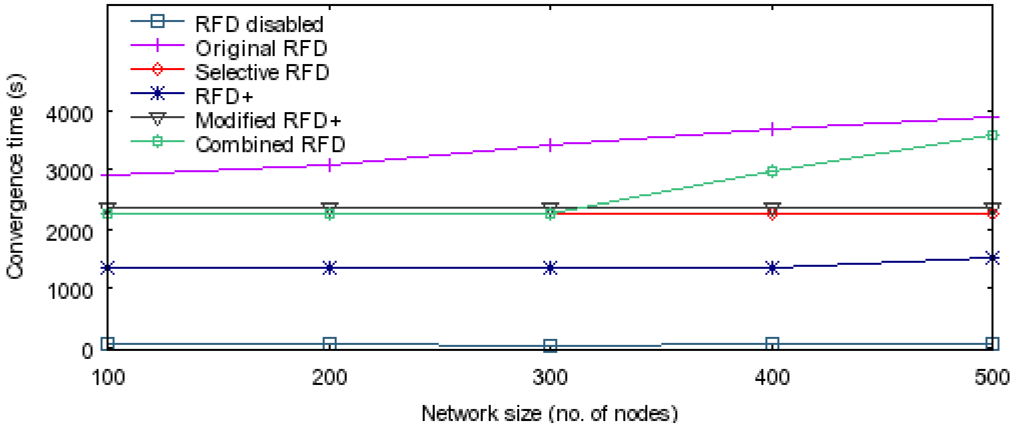

Figure 6.3: BRITE-generated topologies: comparison of convergence time between various RFD algorithms when a route flaps 8 times. Inter-arrival time between updates is 120 s.

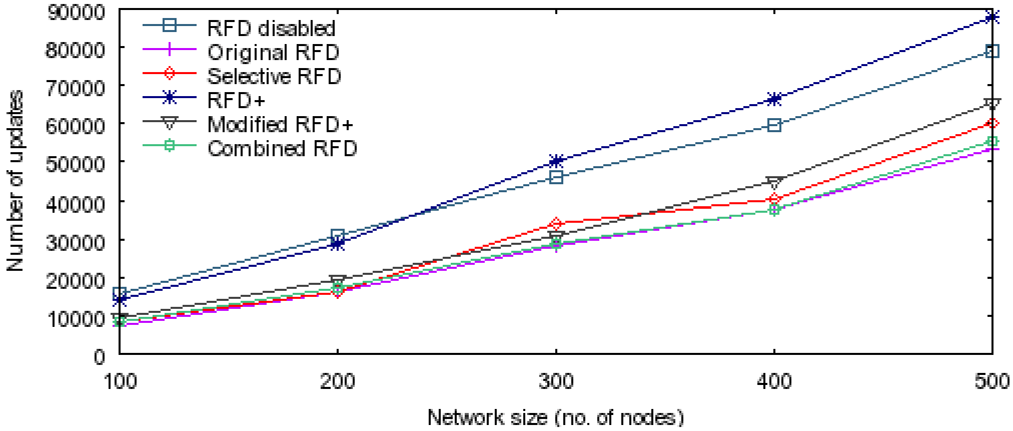

Figure 6.4: BRITE-generated topologies: comparison of number of updates between various RFD algorithms when a route flaps 8 times. Inter-arrival time between updates is 120 s.

In addition to switching between different RFD algorithms, it is also useful to dynamically change certain RFD parameters, such as attribute change penalty and half life. For example, there are occasions when a suppressed route needs to be reused earlier rather than wait for a minimum of ~20 min. An adaptive approach can better suit the dynamics of the Internet routing system.

6.4 RFD and Network Security

The route flap damping mechanism was designed to reduce BGP update messages generated in the network and alleviate the processing load imposed on BGP speakers. However, it may be misused by people with malicious intentions. An example is the exploitation of RFD to cause long isolation between Autonomous Systems (ASs) in the Internet [42]. A malicious attacker may first take advantage of the known vulnerabilities of TCP (Transmission Control Protocol) and ICMP (Internet Control Message Protocol) to attack the underlying TCP connection and cause a BGP peering session to be reset. This causes the affected peers to terminate the current BGP session and then re-establish a new BGP session between them. This implies that withdrawals followed by re-advertisements will be sent between peers regarding their entire routing tables. Depending on the network topology and the number of successfully launched attacks, these withdrawals and re-advertisements may cause extensive route suppressions in the network and result in AS-to-AS or AS-to-prefix isolations when RFD is enabled. RFD is exploited to amplify the adverse effects caused by BGP session attacks, which amounts to a potential denial of service (DOS) attack. A determined attacker can achieve a high probability of AS-to-AS or AS-to-prefix isolation even if the success rate of an individual BGP session attack is low. Some existing techniques, such as *BGP Graceful Restart* (allowing a grace period before withdrawing all the routes when a BGP peering session terminates), can partially mitigate the problem. Nevertheless, we need effective countermeasures to defeat such attacks against the BGP infrastructure and to upset the malicious exploitation of the RFD mechanism. Future RFD implementations may need to take this security aspect of RFD into consideration.

7 CONCLUSIONS

In this thesis, we implemented five RFD algorithms in ns-2. They are based on the ns-BGP and SSFNET BGP-4 modules. O*riginal RFD*, *selective RFD*, and *RFD+* are three existing RFD algorithms. *Modified RFD+* and *combined RFD* are two improvements that we proposed.

We compared the performance of three existing RFD algorithms: *original RFD*, *selective RFD*, and *RFD+*. Simulation results suggested that no algorithm performs optimally in all circumstances. *Original RFD* is more efficient than *selective RFD* and *RFD+* in suppressing persistently flapping routes and achieving network stability. However, *original RFD* may cause significant convergence delay in the case of occasional flaps. *Selective RFD* performs better than *original RFD* in identifying genuine route flaps. However, *selective RFD* does not always identify route flaps correctly. *Selective RFD* may also require the occurrence of additional flaps before suppressing a flapping route, resulting in delayed suppression of persistently flapping routes. This situation worsens when the inter-arrival time between updates increases. *RFD+* has the advantage of reporting no additional flaps. It correctly identifies route flaps in the case of a single flap. Nevertheless, *RFD+* may underestimate the number of genuine route flaps. This delays the suppression of persistently flapping routes. *Selective RFD* and *RFD+* are not as aggressive as *original RFD* in suppressing flapping routes.

We proposed two improvements: *modified RFD+* and *combined RFD*. *Modified RFD+* is a simple modification to *RFD+* and aims to remedy *RFD+*'s problem of underestimating route flaps. *Modified RFD+* can identify route flaps better than other RFD algorithms in both cases of occasional and persistent flaps. *Combined RFD* is a simple adaptive algorithm that integrates *original RFD* and *modified RFD+*. It takes advantage of the strengths of both *original RFD* and *modified RFD+*. *Combined RFD* can efficiently suppress persistent flaps without suppressing a route that flaps only once or twice. In the case of persistent flaps, *combined RFD* also tends to generate fewer update messages than *selective RFD*, *RFD+*, and *modified RFD+*.

An adaptive approach in route flap damping helps achieve a balance between network stability and availability of routes. Future RFD implementations may incorporate algorithms that deal with such issues as unfair suppression, slow convergence, and low level of security.

APPENDIX A TEST SCRIPTS FOR VALIDATION TESTS

The tcl scripts used for RFD validation tests in ns-2 are listed here. These test scripts vary in network topology, simulation duration, and number of simulated route flaps. Four network topologies are adopted: *line* topology (A.1), *tree* topology (A.2), *clique* topology (A.3), and *fork* topology (A.4).

A.1 *Line* Topology

```
puts ""
puts "Route Flap Damping Validation Test 1:"
puts ""
puts "Two ASs, each with one router. The routers are directly
puts "connected, and are each running BGP."
puts ""
puts "       AS 0        AS 1"
puts "        n0 }------{ n1"
puts ""

set ns [new Simulator]

$ns node-config -BGP ON
set n0 [$ns node 0:10.0.0.1]
set n1 [$ns node 1:10.0.1.1]
$ns node-config -BGP OFF

$ns duplex-link $n0 $n1 1Mb 1ms DropTail

set bgp_agent0 [$n0 get-bgp-agent]
$bgp_agent0 bgp-id 10.0.0.1
$bgp_agent0 neighbor 10.0.1.1 remote-as 1
$bgp_agent0 dampening

set bgp_agent1 [$n1 get-bgp-agent]
$bgp_agent1 bgp-id 10.0.1.1
$bgp_agent1 neighbor 10.0.0.1 remote-as 0
$bgp_agent1 dampening

$ns at 0.25 "puts \"\n time: 0.25 \n n0 (ip_addr 10.0.0.1) \
                       defines a network 10.0.2.0/24.\""
$ns at 0.25 "$bgp_agent0 network 10.0.2.0/24"

$ns at 0.26 "puts \"\n time: 0.26 \n n0 (ip_addr 10.0.0.1) \
                       defines a network 10.0.5.0/24.\""
```

```
$ns at 0.26 "$bgp_agent0 network 10.0.5.0/24"

$ns at 0.27 "puts \"\n time: 0.27 \n n0 (ip_addr 10.0.0.1) \
                       defines a network 10.0.6.0/24.\""
$ns at 0.27 "$bgp_agent0 network 10.0.6.0/24"

$ns at 0.3 "puts \"\n time: 0.3 \n n1 (ip_addr 10.0.1.1) \
                       defines a network 10.0.3.0/24.\""
$ns at 0.3 "$bgp_agent1 network 10.0.3.0/24"

$ns at 0.31 "puts \"\n time: 0.31 \n n1 (ip_addr 10.0.1.1) \
                       defines a network 10.0.7.0/24.\""
$ns at 0.31 "$bgp_agent1 network 10.0.7.0/24"

$ns at 0.32 "puts \"\n time: 0.32 \n n1 (ip_addr 10.0.1.1) \
                       defines a network 10.0.8.0/24.\""
$ns at 0.32 "$bgp_agent1 network 10.0.8.0/24"

$ns at 31.0 "puts \"\n time: 31 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 31.0 "$bgp_agent0 show-routes"
$ns at 31.0 "$bgp_agent1 show-routes"

$ns at 31.25 "puts \"\n time: 31.25 \n n0 (ip_addr 10.0.0.1) \
                       withdraws the network 10.0.6.0/24.\""
$ns at 31.25 "$bgp_agent0 no-network 10.0.6.0/24"

$ns at 31.35 "puts \"\n time: 31.35 \n n1 (ip_addr 10.0.1.1) \
                       withdraws the network 10.0.3.0/24.\""
$ns at 31.35 "$bgp_agent1 no-network 10.0.3.0/24"

$ns at 33.0 "puts \"\n time: 33 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 33.0 "$bgp_agent0 show-routes"
$ns at 33.0 "$bgp_agent1 show-routes"

$ns at 62.0 "puts \"\n time: 62.0 \n n1 (ip_addr 10.0.1.1) \
                       advertises the network 10.0.3.0/24.\""
$ns at 62.0  "$bgp_agent1 network 10.0.3.0/24"
$ns at 63.0 "puts \"\n time: 63 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 63.0 "$bgp_agent0 show-routes"
$ns at 63.0 "$bgp_agent1 show-routes"

$ns at 69.0 "puts \"\n time: 69.0 \n n1 (ip_addr 10.0.1.1) \
                       withdraws the network 10.0.3.0/24.\""
$ns at 69.0  "$bgp_agent1 no-network 10.0.3.0/24"
$ns at 70.0 "puts \"\n time: 70 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 70.0 "$bgp_agent0 show-routes"
$ns at 70.0 "$bgp_agent1 show-routes"

$ns at 95.0 "puts \"\n time: 95.0 \n n1 (ip_addr 10.0.1.1) \
                       advertises the network 10.0.3.0/24.\""
$ns at 95.0  "$bgp_agent1 network 10.0.3.0/24"
$ns at 96.0 "puts \"\n time: 96 \
                    \n dump routing tables in all BGP agents: \n\""
```

```
$ns at 96.0 "$bgp_agent0 show-routes"
$ns at 96.0 "$bgp_agent1 show-routes"

$ns at 109.0 "puts \"\n time: 109.0 \n n1 (ip_addr 10.0.1.1) \
                       withdraws the network 10.0.3.0/24.\""
$ns at 109.0  "$bgp_agent1 no-network 10.0.3.0/24"
$ns at 110.0 "puts \"\n time: 110 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 110.0 "$bgp_agent0 show-routes"
$ns at 110.0 "$bgp_agent1 show-routes"

$ns at 126.0 "puts \"\n time: 126.0 \n n1 (ip_addr 10.0.1.1) \
                       advertises the network 10.0.3.0/24.\""
$ns at 126.0  "$bgp_agent1 network 10.0.3.0/24"
$ns at 127.0 "puts \"\n time: 127 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 127.0 "$bgp_agent0 show-routes"
$ns at 127.0 "$bgp_agent1 show-routes"

$ns at 137.0 "puts \"\n time: 137.0 \n n1 (ip_addr 10.0.1.1) \
                       withdraws the network 10.0.3.0/24.\""
$ns at 137.0  "$bgp_agent1 no-network 10.0.3.0/24"
$ns at 158.0 "puts \"\n time: 158.0 \n n1 (ip_addr 10.0.1.1) \
                       advertises the network 10.0.3.0/24.\""
$ns at 158.0  "$bgp_agent1 network 10.0.3.0/24"

#Note that route 10.0.3.0/24 is suppressed
$ns at 160.0 "puts \"\n time: 160 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 160.0 "$bgp_agent0 show-routes"
$ns at 160.0 "$bgp_agent0 show-all"
$ns at 160.0 "$bgp_agent1 show-routes"
$ns at 160.0 "$bgp_agent1 show-all"

$ns at 1500.0 "puts \"\n time: 1500 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 1500.0 "$bgp_agent0 show-routes"
$ns at 1500.0 "$bgp_agent0 show-all"
$ns at 1500.0 "$bgp_agent1 show-routes"
$ns at 1500.0 "$bgp_agent1 show-all"

#Note that route 10.0.3.0/24 is released
$ns at 3000.0 "puts \"\n time: 3000 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 3000.0 "$bgp_agent0 show-routes"
$ns at 3000.0 "$bgp_agent0 show-all"
$ns at 3000.0 "$bgp_agent0 show-damping"
$ns at 3000.0 "$bgp_agent1 show-routes"
$ns at 3000.0 "$bgp_agent1 show-all"
$ns at 3000.0 "$bgp_agent1 show-damping"

$ns at 3100.0 "finish"

proc finish {} {
      global ns
      puts "Simulation finished. "
      exit 0
}

puts "Simulation starts..."
$ns run
```

A.2 *Tree* Topology

```
puts ""
puts "Route Flap Damping Validation Test 2:"
puts ""
puts "Four ASs, each with one router. The routers are connected in a"
puts "tree-like fashion, and are each running BGP."
puts ""
puts "                    AS 3  "
puts "                   { n3 } "
puts "                     | "
puts "                     | "
puts "                     | "
puts "                    AS 0            "
puts "                   { n0 }           "
puts "                  |      |          "
puts "                 |        |         "
puts "                |          |        "
puts "               |            |       "
puts "            { n2 }        { n1 }    "
puts "             AS 2          AS 1     "

set ns [new Simulator]

$ns node-config -BGP ON
set n0 [$ns node 0:10.0.0.4]
set n1 [$ns node 1:10.0.1.3]
set n2 [$ns node 2:10.0.2.2]
set n3 [$ns node 3:10.0.3.1]
$ns node-config -BGP OFF

$ns duplex-link $n0 $n1 1Mb 1ms DropTail
$ns duplex-link $n2 $n0 1Mb 1ms DropTail
$ns duplex-link $n0 $n3 1Mb 1ms DropTail

set bgp_agent0 [$n0 get-bgp-agent]
$bgp_agent0 bgp-id 10.0.0.4
$bgp_agent0 neighbor 10.0.1.3 remote-as 1
$bgp_agent0 neighbor 10.0.2.2 remote-as 2
$bgp_agent0 neighbor 10.0.3.1 remote-as 3
$bgp_agent0 dampening 0 0 3000 750 900 1000 500 3600

set bgp_agent1 [$n1 get-bgp-agent]
$bgp_agent1 bgp-id 10.0.1.3
$bgp_agent1 neighbor 10.0.0.4 remote-as 0
$bgp_agent1 dampening 0 0 3000 750 900 1000 500 3600

set bgp_agent2 [$n2 get-bgp-agent]
$bgp_agent2 bgp-id 10.0.2.2
$bgp_agent2 neighbor 10.0.0.4 remote-as 0
```

```
$bgp_agent2 dampening 0 0 3000 750 900 1000 500 3600

set bgp_agent3 [$n3 get-bgp-agent]
$bgp_agent3 bgp-id 10.0.3.1
$bgp_agent3 neighbor 10.0.0.4 remote-as 0
$bgp_agent3 dampening 0 0 3000 750 900 1000 500 3600

$ns at 0.2 "puts \"\n time: 0.2 \n n0 (ip_addr 10.0.0.4) \
                       defines a network 10.0.4.0/24.\""
$ns at 0.2 "$bgp_agent0 network 10.0.4.0/24"

$ns at 2.22 "puts \"\n time: 2.22 \n n0 (ip_addr 10.0.0.4) \
                       defines a network 10.0.5.0/24.\""
$ns at 2.22 "$bgp_agent0 network 10.0.5.0/24"

$ns at 2.57 "puts \"\n time: 2.57 \n n0 (ip_addr 10.0.0.4) \
                       defines a network 10.0.6.0/24.\""
$ns at 2.57 "$bgp_agent0 network 10.0.6.0/24"

$ns at 3.3 "puts \"\n time: 3.3 \n n1 (ip_addr 10.0.1.3) \
                       defines a network 10.0.7.0/24.\""
$ns at 3.3 "$bgp_agent1 network 10.0.7.0/24"

$ns at 4.31 "puts \"\n time: 4.31 \n n1 (ip_addr 10.0.1.3) \
                       defines a network 10.0.8.0/24.\""
$ns at 4.31 "$bgp_agent1 network 10.0.8.0/24"

$ns at 5.32 "puts \"\n time: 5.32 \n n1 (ip_addr 10.0.1.3) \
                       defines a network 10.0.9.0/24.\""
$ns at 5.32 "$bgp_agent1 network 10.0.9.0/24"

$ns at 6.35 "puts \"\n time: 6.35 \n n2 (ip_addr 10.0.2.2) \
                       defines a network 10.1.7.0/24.\""
$ns at 6.35 "$bgp_agent2 network 10.1.7.0/24"

$ns at 7.37 "puts \"\n time: 7.37 \n n2 (ip_addr 10.0.2.2) \
                       defines a network 10.1.8.0/24.\""
$ns at 7.37 "$bgp_agent2 network 10.1.8.0/24"

$ns at 62.0 "puts \"\n time: 62 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 62.0 "$bgp_agent0 show-routes"
$ns at 62.0 "$bgp_agent0 show-all"
$ns at 62.0 "$bgp_agent1 show-routes"
$ns at 62.0 "$bgp_agent1 show-all"
$ns at 62.0 "$bgp_agent2 show-routes"
$ns at 62.0 "$bgp_agent2 show-all"
$ns at 62.0 "$bgp_agent3 show-routes"
$ns at 62.0 "$bgp_agent3 show-all"

$ns at 62.05 "puts \"\n time: 62.05 \n n2 (ip_addr 10.0.2.2) \
                       withdraws the network 10.1.7.0/24.\""
$ns at 62.05 "$bgp_agent2 no-network 10.1.7.0/24"

$ns at 62.85 "puts \"\n time: 62.85 \n n1 (ip_addr 10.0.1.3) \
                       withdraws the network 10.0.7.0/24.\""
$ns at 62.85 "$bgp_agent1 no-network 10.0.7.0/24"
```

```
$ns at 66.0 "puts \"\n time: 66 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 66.0 "$bgp_agent0 show-routes"
$ns at 66.0 "$bgp_agent0 show-all"
$ns at 66.0 "$bgp_agent1 show-routes"
$ns at 66.0 "$bgp_agent1 show-all"
$ns at 66.0 "$bgp_agent2 show-routes"
$ns at 66.0 "$bgp_agent2 show-all"
$ns at 66.0 "$bgp_agent3 show-routes"
$ns at 66.0 "$bgp_agent3 show-all"

$ns at 95.0 "puts \"\n time: 95.0 \n n1 (ip_addr 10.0.1.3) \
                       advertises the network 10.0.7.0/24.\""
$ns at 95.0  "$bgp_agent1 network 10.0.7.0/24"
$ns at 100.0 "puts \"\n time: 100 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 100.0 "$bgp_agent0 show-routes"
$ns at 100.0  "$bgp_agent0 show-all"
$ns at 100.0 "$bgp_agent1 show-routes"
$ns at 100.0 "$bgp_agent1 show-all"
$ns at 100.0 "$bgp_agent2 show-routes"
$ns at 100.0 "$bgp_agent2 show-all"
$ns at 100.0 "$bgp_agent3 show-routes"
$ns at 100.0 "$bgp_agent3 show-all"

$ns at 200.0 "puts \"\n time: 200.0 \n n1 (ip_addr 10.0.1.3) \
                       advertises the network 10.0.7.0/24.\""
$ns at 200.0  "$bgp_agent1 network 10.0.7.0/24"
$ns at 210.0 "puts \"\n time: 210 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 210.0 "$bgp_agent0 show-routes"
$ns at 210.0 "$bgp_agent0 show-all"
$ns at 210.0 "$bgp_agent0 show-damping"
$ns at 210.0 "$bgp_agent1 show-routes"
$ns at 210.0 "$bgp_agent1 show-all"
$ns at 210.0 "$bgp_agent1 show-damping"
$ns at 210.0 "$bgp_agent2 show-routes"
$ns at 210.0 "$bgp_agent2 show-all"
$ns at 210.0 "$bgp_agent2 show-damping"
$ns at 210.0 "$bgp_agent3 show-routes"
$ns at 210.0 "$bgp_agent3 show-all"
$ns at 210.0 "$bgp_agent3 show-damping"

$ns at 250.0 "puts \"\n time: 250.0 \n n1 (ip_addr 10.0.1.3) \
                       withdraws the network 10.0.7.0/24.\""
$ns at 250.0  "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 260.0 "puts \"\n time: 260 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 260.0 "$bgp_agent0 show-routes"
$ns at 260.0  "$bgp_agent0 show-all"
$ns at 260.0 "$bgp_agent1 show-routes"
$ns at 260.0 "$bgp_agent1 show-all"
$ns at 260.0 "$bgp_agent2 show-routes"
$ns at 260.0 "$bgp_agent2 show-all"
$ns at 260.0 "$bgp_agent3 show-routes"
$ns at 260.0 "$bgp_agent3 show-all"
```

```
$ns at 300.0 "puts \"\n time: 300.0 \n n1 (ip_addr 10.0.1.3) \
                       advertises the network 10.0.7.0/24.\""
$ns at 300.0  "$bgp_agent1 network 10.0.7.0/24"
$ns at 340.0 "puts \"\n time: 340 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 340.0 "$bgp_agent0 show-routes"
$ns at 340.0 "$bgp_agent0 show-all"
$ns at 340.0 "$bgp_agent0 show-damping"
$ns at 340.0 "$bgp_agent1 show-routes"
$ns at 340.0 "$bgp_agent1 show-all"
$ns at 340.0 "$bgp_agent1 show-damping"
$ns at 340.0 "$bgp_agent2 show-routes"
$ns at 340.0 "$bgp_agent2 show-all"
$ns at 340.0 "$bgp_agent2 show-damping"
$ns at 340.0 "$bgp_agent3 show-routes"
$ns at 340.0 "$bgp_agent3 show-all"
$ns at 340.0 "$bgp_agent3 show-damping"

$ns at 350.0 "puts \"\n time: 350.0 \n n1 (ip_addr 10.0.1.3) \
                   begins to have a series of advertisements and \
                   withdrawals regarding the network 10.0.7.0/24.\""
$ns at 350.0  "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 400.0  "$bgp_agent1 network 10.0.7.0/24"
$ns at 450.0  "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 500.0  "$bgp_agent1 network 10.0.7.0/24"
$ns at 550.0  "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 600.0  "$bgp_agent1 network 10.0.7.0/24"
$ns at 650.0  "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 700.0  "$bgp_agent1 network 10.0.7.0/24"
$ns at 750.0  "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 800.0  "$bgp_agent1 network 10.0.7.0/24"
$ns at 850.0 "puts \"\n time: 850 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 850.0 "$bgp_agent0 show-routes"
$ns at 850.0 "$bgp_agent0 show-all"
$ns at 850.0 "$bgp_agent0 show-damping"
$ns at 850.0 "$bgp_agent1 show-routes"
$ns at 850.0 "$bgp_agent1 show-all"
$ns at 850.0 "$bgp_agent1 show-damping"
$ns at 850.0 "$bgp_agent2 show-routes"
$ns at 850.0 "$bgp_agent2 show-all"
$ns at 850.0 "$bgp_agent2 show-damping"
$ns at 850.0 "$bgp_agent3 show-routes"
$ns at 850.0 "$bgp_agent3 show-all"
$ns at 850.0 "$bgp_agent3 show-damping"

$ns at 1200.0 "puts \"\n time: 1200.0 \n n2 (ip_addr 10.0.2.2) \
                   begins to have a series of advertisements and \
                   withdrawals regarding the network 10.1.7.0/24.\""
$ns at  1200.0 "$bgp_agent2 network 10.1.7.0/24"
$ns at  1300.0 "$bgp_agent2 no-network 10.1.7.0/24"
$ns at  1400.0 "$bgp_agent2 network 10.1.7.0/24"
$ns at  1500.0 "$bgp_agent2 no-network 10.1.7.0/24"
$ns at  1600.0 "$bgp_agent2 network 10.1.7.0/24"
$ns at  1700.0 "$bgp_agent2 no-network 10.1.7.0/24"
$ns at  1800.0 "$bgp_agent2 network 10.1.7.0/24"
```

```
$ns at  1900.0 "$bgp_agent2 no-network 10.1.7.0/24"
$ns at  2000.0 "$bgp_agent2 network 10.1.7.0/24"

$ns at 2005.0 "puts \"\n time: 2005 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 2005.0 "$bgp_agent0 show-routes"
$ns at 2005.0  "$bgp_agent0 show-all"
$ns at 2005.0 "$bgp_agent0 show-damping"
$ns at 2005.0 "$bgp_agent1 show-routes"
$ns at 2005.0  "$bgp_agent1 show-all"
$ns at 2005.0 "$bgp_agent1 show-damping"
$ns at 2005.0 "$bgp_agent2 show-routes"
$ns at 2005.0  "$bgp_agent2 show-all"
$ns at 2005.0 "$bgp_agent2 show-damping"
$ns at 2005.0 "$bgp_agent3 show-routes"
$ns at 2005.0  "$bgp_agent3 show-all"
$ns at 2005.0 "$bgp_agent3 show-damping"

$ns at 3500.0 "puts \"\n time: 3500 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 3500.0 "$bgp_agent0 show-routes"
$ns at 3500.0  "$bgp_agent0 show-all"
$ns at 3500.0 "$bgp_agent0 show-damping"
$ns at 3500.0 "$bgp_agent1 show-routes"
$ns at 3500.0  "$bgp_agent1 show-all"
$ns at 3500.0 "$bgp_agent1 show-damping"
$ns at 3500.0 "$bgp_agent2 show-routes"
$ns at 3500.0  "$bgp_agent2 show-all"
$ns at 3500.0 "$bgp_agent2 show-damping"
$ns at 3500.0 "$bgp_agent3 show-routes"
$ns at 3500.0  "$bgp_agent3 show-all"
$ns at 3500.0 "$bgp_agent3 show-damping"

$ns at 3530.0 "puts \"\n time: 3530.0 \n n1 (ip_addr 10.0.1.3) \
                       advertises the network 10.0.7.0/24.\""
$ns at 3530.0  "$bgp_agent1 network 10.0.7.0/24"
$ns at 3535.0 "puts \"\n time: 3535.0 \n n2 (ip_addr 10.0.2.2) \
                       advertises the network 10.1.7.0/24.\""
$ns at 3535.0  "$bgp_agent2 network 10.1.7.0/24"

$ns at 3550.0 "puts \"\n time: 3550 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 3550.0 "$bgp_agent0 show-routes"
$ns at 3550.0  "$bgp_agent0 show-all"
$ns at 3550.0 "$bgp_agent1 show-routes"
$ns at 3550.0  "$bgp_agent1 show-all"
$ns at 3550.0 "$bgp_agent2 show-routes"
$ns at 3550.0  "$bgp_agent2 show-all"
$ns at 3550.0 "$bgp_agent3 show-routes"
$ns at 3550.0  "$bgp_agent3 show-all"

$ns at 3580.0 "puts \"\n time: 3580.0 \n n1 (ip_addr 10.0.1.3) \
                       withdraws the network 10.0.7.0/24.\""
$ns at 3580.0  "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 3600.0 "puts \"\n time: 3600 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 3600.0 "$bgp_agent0 show-routes"
```

```
$ns at 3600.0  "$bgp_agent0 show-all"
$ns at 3600.0 "$bgp_agent1 show-routes"
$ns at 3600.0  "$bgp_agent1 show-all"
$ns at 3600.0 "$bgp_agent2 show-routes"
$ns at 3600.0  "$bgp_agent2 show-all"
$ns at 3600.0 "$bgp_agent3 show-routes"
$ns at 3600.0  "$bgp_agent3 show-all"

$ns at 3650.0 "puts \"\n time: 3650.0 \n n1 (ip_addr 10.0.1.3) \
                       advertises the network 10.0.7.0/24.\""
$ns at 3650.0  "$bgp_agent1 network 10.0.7.0/24"
$ns at 3700.0 "puts \"\n time: 3700 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 3700.0 "$bgp_agent0 show-routes"
$ns at 3700.0  "$bgp_agent0 show-all"
$ns at 3700.0 "$bgp_agent1 show-routes"
$ns at 3700.0  "$bgp_agent1 show-all"
$ns at 3700.0 "$bgp_agent2 show-routes"
$ns at 3700.0  "$bgp_agent2 show-all"
$ns at 3700.0 "$bgp_agent3 show-routes"
$ns at 3700.0  "$bgp_agent3 show-all"

$ns at 3750.0 "puts \"\n time: 3750.0 \n n1 (ip_addr 10.0.1.3) \
                       withdraws the network 10.0.7.0/24.\""
$ns at 3750.0  "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 3800.0 "puts \"\n time: 3800 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 3800.0 "$bgp_agent0 show-routes"
$ns at 3800.0  "$bgp_agent0 show-all"
$ns at 3800.0 "$bgp_agent1 show-routes"
$ns at 3800.0  "$bgp_agent1 show-all"
$ns at 3800.0 "$bgp_agent2 show-routes"
$ns at 3800.0  "$bgp_agent2 show-all"
$ns at 3800.0 "$bgp_agent3 show-routes"
$ns at 3800.0  "$bgp_agent3 show-all"

$ns at 3830.0 "puts \"\n time: 3830.0 \n n1 (ip_addr 10.0.1.3) \
                   begins to have a series of advertisements and \
                   withdrawals regarding the network 10.0.7.0/24.\""
$ns at 3830.0  "$bgp_agent1 network 10.0.7.0/24"
$ns at 3840.0  "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 3870.0  "$bgp_agent1 network 10.0.7.0/24"
$ns at 3880.0  "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 3910.0  "$bgp_agent1 network 10.0.7.0/24"
$ns at 3920.0  "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 3950.0  "$bgp_agent1 network 10.0.7.0/24"
$ns at 3960.0  "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 3990.0  "$bgp_agent1 network 10.0.7.0/24"

$ns at 4005.0 "puts \"\n time: 4005 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 4005.0 "$bgp_agent0 show-routes"
$ns at 4005.0  "$bgp_agent0 show-all"
$ns at 4005.0 "$bgp_agent0 show-damping"
$ns at 4005.0 "$bgp_agent1 show-routes"
$ns at 4005.0  "$bgp_agent1 show-all"
$ns at 4005.0 "$bgp_agent1 show-damping"
```

```
$ns at 4005.0 "$bgp_agent2 show-routes"
$ns at 4005.0  "$bgp_agent2 show-all"
$ns at 4005.0 "$bgp_agent2 show-damping"
$ns at 4005.0 "$bgp_agent3 show-routes"
$ns at 4005.0  "$bgp_agent3 show-all"
$ns at 4005.0 "$bgp_agent3 show-damping"

$ns at 6005.0 "puts \"\n time: 6005 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 6005.0 "$bgp_agent0 show-routes"
$ns at 6005.0  "$bgp_agent0 show-all"
$ns at 6005.0 "$bgp_agent1 show-routes"
$ns at 6005.0  "$bgp_agent1 show-all"
$ns at 6005.0 "$bgp_agent2 show-routes"
$ns at 6005.0  "$bgp_agent2 show-all"
$ns at 6005.0 "$bgp_agent3 show-routes"
$ns at 6005.0  "$bgp_agent3 show-all"

$ns at 8050.0 "puts \"\n time: 8050 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 8050.0 "$bgp_agent0 show-routes"
$ns at 8050.0  "$bgp_agent0 show-all"
$ns at 8050.0 "$bgp_agent0 show-damping"
$ns at 8050.0 "$bgp_agent1 show-routes"
$ns at 8050.0  "$bgp_agent1 show-all"
$ns at 8050.0 "$bgp_agent1 show-damping"
$ns at 8050.0 "$bgp_agent2 show-routes"
$ns at 8050.0  "$bgp_agent2 show-all"
$ns at 8050.0 "$bgp_agent2 show-damping"
$ns at 8050.0 "$bgp_agent3 show-routes"
$ns at 8050.0  "$bgp_agent3 show-all"
$ns at 8050.0 "$bgp_agent3 show-damping"

$ns at 8100.0 "puts \"\n time: 8100.0 \n n1 (ip_addr 10.0.1.3) \
                   begins to have a series of advertisements and \
                   withdrawals regarding the network 10.0.7.0/24.\""
$ns at 8100.0 "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 8120.0 "$bgp_agent1 network 10.0.7.0/24"
$ns at 8140.0 "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 8160.0 "$bgp_agent1 network 10.0.7.0/24"
$ns at 8180.0 "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 8200.0 "$bgp_agent1 network 10.0.7.0/24"
$ns at 8220.0 "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 8240.0 "$bgp_agent1 network 10.0.7.0/24"
$ns at 8260.0 "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 8280.0 "$bgp_agent1 network 10.0.7.0/24"
$ns at 8300.0 "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 8320.0 "$bgp_agent1 network 10.0.7.0/24"
$ns at 8340.0 "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 8360.0 "$bgp_agent1 network 10.0.7.0/24"
$ns at 8380.0 "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 8400.0 "$bgp_agent1 network 10.0.7.0/24"
$ns at 8420.0 "$bgp_agent1 network 10.0.7.0/24"
$ns at 8430.0 "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 8460.0 "$bgp_agent1 network 10.0.7.0/24"
$ns at 8480.0 "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 8500.0 "$bgp_agent1 network 10.0.7.0/24"
```

```
$ns at 8520.0 "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 8535.0 "$bgp_agent1 network 10.0.7.0/24"
$ns at 8550.0 "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 8570.0 "$bgp_agent1 network 10.0.7.0/24"
$ns at 8580.0 "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 8605.0 "$bgp_agent1 network 10.0.7.0/24"
$ns at 8610.0 "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 8640.0 "$bgp_agent1 network 10.0.7.0/24"
$ns at 8650.0 "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 8675.0 "$bgp_agent1 network 10.0.7.0/24"
$ns at 8690.0 "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 8710.0 "$bgp_agent1 network 10.0.7.0/24"
$ns at 8720.0 "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 8750.0 "$bgp_agent1 network 10.0.7.0/24"
$ns at 8760.0 "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 8785.0 "$bgp_agent1 network 10.0.7.0/24"
$ns at 8800.0 "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 8820.0 "$bgp_agent1 network 10.0.7.0/24"
$ns at 8830.0 "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 8855.0 "$bgp_agent1 network 10.0.7.0/24"
$ns at 8870.0 "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 8890.0 "$bgp_agent1 network 10.0.7.0/24"

$ns at 8900.0 "puts \"\n time: 8900 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 8900.0 "$bgp_agent0 show-routes"
$ns at 8900.0  "$bgp_agent0 show-all"
$ns at 8900.0 "$bgp_agent0 show-damping"
$ns at 8900.0 "$bgp_agent1 show-routes"
$ns at 8900.0  "$bgp_agent1 show-all"
$ns at 8900.0 "$bgp_agent1 show-damping"
$ns at 8900.0 "$bgp_agent2 show-routes"
$ns at 8900.0  "$bgp_agent2 show-all"
$ns at 8900.0 "$bgp_agent2 show-damping"
$ns at 8900.0 "$bgp_agent3 show-routes"
$ns at 8900.0  "$bgp_agent3 show-all"
$ns at 8900.0 "$bgp_agent3 show-damping"

$ns at 12500.0 "puts \"\n time: 12500 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 12500.0 "$bgp_agent0 show-routes"
$ns at 12500.0  "$bgp_agent0 show-all"
$ns at 12500.0 "$bgp_agent0 show-damping"
$ns at 12500.0 "$bgp_agent1 show-routes"
$ns at 12500.0  "$bgp_agent1 show-all"
$ns at 12500.0 "$bgp_agent1 show-damping"
$ns at 12500.0 "$bgp_agent2 show-routes"
$ns at 12500.0  "$bgp_agent2 show-all"
$ns at 12500.0 "$bgp_agent2 show-damping"
$ns at 12500.0 "$bgp_agent3 show-routes"
$ns at 12500.0  "$bgp_agent3 show-all"
$ns at 12500.0 "$bgp_agent3 show-damping"

$ns at 12580.0 "puts \"\n time: 12580.0 \n n1 (ip_addr 10.0.1.3) \
                       withdraws the network 10.0.7.0/24.\""
$ns at 12580.0 "$bgp_agent1 no-network 10.0.7.0/24"
$ns at 12590.0 "puts \"\n time: 12590 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 12590.0 "$bgp_agent0 show-routes"
$ns at 12590.0  "$bgp_agent0 show-all"
$ns at 12590.0 "$bgp_agent1 show-routes"
$ns at 12590.0  "$bgp_agent1 show-all"
$ns at 12590.0 "$bgp_agent2 show-routes"
$ns at 12590.0  "$bgp_agent2 show-all"
$ns at 12590.0 "$bgp_agent3 show-routes"
$ns at 12590.0  "$bgp_agent3 show-all"

$ns at 12600.0 "puts \"\n time: 12600.0 \n n1 (ip_addr 10.0.1.3) \
                       advertises the network 10.0.7.0/24.\""
$ns at 12600.0 "$bgp_agent1 network 10.0.7.0/24"
$ns at 12610.0 "puts \"\n time: 12610 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 12610.0 "$bgp_agent0 show-routes"
$ns at 12610.0  "$bgp_agent0 show-all"
$ns at 12610.0 "$bgp_agent0 show-damping"
$ns at 12610.0 "$bgp_agent1 show-routes"
$ns at 12610.0  "$bgp_agent1 show-all"
$ns at 12610.0 "$bgp_agent1 show-damping"
$ns at 12610.0 "$bgp_agent2 show-routes"
$ns at 12610.0  "$bgp_agent2 show-all"
$ns at 12610.0 "$bgp_agent2 show-damping"
$ns at 12610.0 "$bgp_agent3 show-routes"
$ns at 12610.0  "$bgp_agent3 show-all"
$ns at 12610.0 "$bgp_agent3 show-damping"

$ns at 12800.0 "finish"

proc finish {} {
      global ns
      puts "Simulation finished. "
      exit 0
}

puts "Simulation starts..."
$ns run
```

A.3 *Clique* Topology

```
puts ""
puts "Route Flap Damping Validation Test 3:"
puts ""
puts "Seven ASs, each with one router.  The routers are connected"
puts "in the following way, and are each running BGP."
puts ""
puts "                   -------------            "
puts "                  /n2          n3\          "
puts "                 /                \         "
puts "                /                  \        "
puts "               /                    \       "
puts "              /       a clique       \      "
puts "              \n1     of size 5    n4/      "
puts "               \                    /       "
puts "                \                  /        "
puts "                 \                /         "
puts "                  \      n5      /          "
puts "                   -------------/----n0     "
puts "                                            "

set ns [new Simulator]

$ns node-config -BGP ON
set n0 [$ns node 0:10.0.0.1]
set n1 [$ns node 1:10.0.1.1]
set n2 [$ns node 2:10.0.2.1]
set n3 [$ns node 3:10.0.3.1]
set n4 [$ns node 4:10.0.4.1]
set n5 [$ns node 5:10.0.5.1]
$ns node-config -BGP OFF

$ns duplex-link $n1 $n2 1Mb 1ms DropTail
$ns duplex-link $n1 $n3 1Mb 1ms DropTail
$ns duplex-link $n1 $n4 1Mb 1ms DropTail
$ns duplex-link $n1 $n5 1Mb 1ms DropTail
$ns duplex-link $n2 $n3 1Mb 1ms DropTail
$ns duplex-link $n2 $n4 1Mb 1ms DropTail
$ns duplex-link $n2 $n5 1Mb 1ms DropTail
$ns duplex-link $n3 $n4 1Mb 1ms DropTail
$ns duplex-link $n3 $n5 1Mb 1ms DropTail
$ns duplex-link $n4 $n5 1Mb 1ms DropTail
$ns duplex-link $n5 $n0 1Mb 1ms DropTail

set bgp_agent0 [$n0 get-bgp-agent]
$bgp_agent0 bgp-id 10.0.0.1
$bgp_agent0 neighbor 10.0.5.1 remote-as 5
$bgp_agent0 dampening 2 0 3000 750 900 1000 500 3600

set bgp_agent1 [$n1 get-bgp-agent]
$bgp_agent1 bgp-id 10.0.1.1
$bgp_agent1 neighbor 10.0.2.1 remote-as 2
$bgp_agent1 neighbor 10.0.3.1 remote-as 3
$bgp_agent1 neighbor 10.0.4.1 remote-as 4
$bgp_agent1 neighbor 10.0.5.1 remote-as 5
$bgp_agent1 dampening 2 0 3000 750 900 1000 500 3600

set bgp_agent2 [$n2 get-bgp-agent]
$bgp_agent2 bgp-id 10.0.2.1
$bgp_agent2 neighbor 10.0.1.1 remote-as 1
$bgp_agent2 neighbor 10.0.3.1 remote-as 3
$bgp_agent2 neighbor 10.0.4.1 remote-as 4
$bgp_agent2 neighbor 10.0.5.1 remote-as 5
$bgp_agent2 dampening 2 0 3000 750 900 1000 500 3600

set bgp_agent3 [$n3 get-bgp-agent]
$bgp_agent3 bgp-id 10.0.3.1
$bgp_agent3 neighbor 10.0.1.1 remote-as 1
$bgp_agent3 neighbor 10.0.2.1 remote-as 2
$bgp_agent3 neighbor 10.0.4.1 remote-as 4
$bgp_agent3 neighbor 10.0.5.1 remote-as 5
$bgp_agent3 dampening 2 0 3000 750 900 1000 500 3600
```

```
set bgp_agent4 [$n4 get-bgp-agent]
$bgp_agent4 bgp-id 10.0.4.1
$bgp_agent4 neighbor 10.0.1.1 remote-as 1
$bgp_agent4 neighbor 10.0.2.1 remote-as 2
$bgp_agent4 neighbor 10.0.3.1 remote-as 3
$bgp_agent4 neighbor 10.0.5.1 remote-as 5
$bgp_agent4 dampening 2 0 3000 750 900 1000 500 3600

set bgp_agent5 [$n5 get-bgp-agent]
$bgp_agent5 bgp-id 10.0.5.1
$bgp_agent5 neighbor 10.0.1.1 remote-as 1
$bgp_agent5 neighbor 10.0.2.1 remote-as 2
$bgp_agent5 neighbor 10.0.3.1 remote-as 3
$bgp_agent5 neighbor 10.0.4.1 remote-as 4
$bgp_agent5 neighbor 10.0.0.1 remote-as 0
$bgp_agent5 dampening 2 0 3000 750 900 1000 500 3600

$ns at 1.25 "puts \"\n time: 1.25 \n n1 (ip_addr 10.0.1.1) \
                       defines a network 10.1.2.0/24.\""
$ns at 1.25 "$bgp_agent1 network 10.1.2.0/24"

$ns at 100.0 "puts \"\n time: 100 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 100.0 "$bgp_agent0 show-routes"
$ns at 100.0 "$bgp_agent0 show-all"
$ns at 100.0 "$bgp_agent1 show-routes"
$ns at 100.0 "$bgp_agent1 show-all"
$ns at 100.0 "$bgp_agent2 show-routes"
$ns at 100.0 "$bgp_agent2 show-all"
$ns at 100.0 "$bgp_agent3 show-routes"
$ns at 100.0 "$bgp_agent3 show-all"
$ns at 100.0 "$bgp_agent4 show-routes"
$ns at 100.0 "$bgp_agent4 show-all"
$ns at 100.0 "$bgp_agent5 show-routes"
$ns at 100.0 "$bgp_agent5 show-all"

$ns at 110 "puts \"\n time: 110 \n n1 (ip_addr 10.0.1.1) \
                       withdraws the network 10.1.2.0/24.\""
$ns at 110 "$bgp_agent1 no-network 10.1.2.0/24"

$ns at 200.0 "puts \"\n time: 200 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 200.0 "$bgp_agent0 show-routes"
$ns at 200.0 "$bgp_agent0 show-all"
$ns at 200.0 "$bgp_agent1 show-routes"
$ns at 200.0 "$bgp_agent1 show-all"
$ns at 200.0 "$bgp_agent2 show-routes"
$ns at 200.0 "$bgp_agent2 show-all"
$ns at 200.0 "$bgp_agent3 show-routes"
$ns at 200.0 "$bgp_agent3 show-all"
$ns at 200.0 "$bgp_agent4 show-routes"
$ns at 200.0 "$bgp_agent4 show-all"
$ns at 200.0 "$bgp_agent5 show-routes"
$ns at 200.0 "$bgp_agent5 show-all"

$ns at 700.0 "$bgp_agent1 network 10.1.2.0/24"
```

```
$ns at 900.0 "puts \"\n time: 900 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 900.0 "$bgp_agent0 show-routes"
$ns at 900.0 "$bgp_agent0 show-all"
$ns at 900.0 "$bgp_agent1 show-routes"
$ns at 900.0 "$bgp_agent1 show-all"
$ns at 900.0 "$bgp_agent2 show-routes"
$ns at 900.0 "$bgp_agent2 show-all"
$ns at 900.0 "$bgp_agent3 show-routes"
$ns at 900.0 "$bgp_agent3 show-all"
$ns at 900.0 "$bgp_agent4 show-routes"
$ns at 900.0 "$bgp_agent4 show-all"
$ns at 900.0 "$bgp_agent5 show-routes"
$ns at 900.0 "$bgp_agent5 show-all"

$ns at 905.0 "$bgp_agent0 show-damping"
$ns at 905.0 "$bgp_agent1 show-damping"
$ns at 905.0 "$bgp_agent2 show-damping"
$ns at 905.0 "$bgp_agent3 show-damping"
$ns at 905.0 "$bgp_agent4 show-damping"
$ns at 905.0 "$bgp_agent5 show-damping"

$ns at 910.0 "finish"

proc finish {} {
      global ns
      puts "Simulation finished. "
      exit 0
}

puts "Simulation starts..."
$ns run
```

A.4 *Fork* Topology

```
puts ""
puts "Route Flap Damping Validation Test 4:"
puts ""
puts "Eleven ASs, each with one router.  The routers are connected"
puts " in the following way, and are each running BGP."
puts ""
puts "               AS9                         "
puts "                n9                         "
puts "                |                          "
puts "                |                          "
puts "       AS5     AS1    AS3     AS4      AS2 "
puts "       n5-------n1-----n3------n4------n2  "
puts "       |        \      |       /       /   "
puts "       |         \     |      /       /    "
puts "       |          \    |     /       /     "
puts "       |           \   |    /       /      "
puts "       |            \  |   /       /       "
puts "       |             \ |  /       /        "
puts "       n6---n7---n8----n0---------         "
puts "      AS6  AS7  AS8   AS0                  "
puts "                       |                   "
puts "                       |                   "
puts "                      n10                  "
puts "                      AS10                 "

set ns [new Simulator]

$ns node-config -BGP ON
set n0 [$ns node 0:10.0.0.1]
set n1 [$ns node 1:10.0.1.1]
set n2 [$ns node 2:10.0.2.1]
set n3 [$ns node 3:10.0.3.1]
set n4 [$ns node 4:10.0.4.1]
set n5 [$ns node 5:10.0.5.1]
set n6 [$ns node 6:10.0.6.1]
set n7 [$ns node 7:10.0.7.1]
set n8 [$ns node 8:10.0.8.1]
set n9 [$ns node 9:10.0.9.1]
set n10 [$ns node 10:10.0.10.1]
$ns node-config -BGP OFF

$ns duplex-link $n0 $n1 1Mb 1ms DropTail
$ns duplex-link $n0 $n3 1Mb 1ms DropTail
$ns duplex-link $n0 $n4 1Mb 1ms DropTail
$ns duplex-link $n0 $n2 1Mb 1ms DropTail
$ns duplex-link $n5 $n1 1Mb 1ms DropTail
$ns duplex-link $n1 $n3 1Mb 1ms DropTail
$ns duplex-link $n3 $n4 1Mb 1ms DropTail
$ns duplex-link $n4 $n2 1Mb 1ms DropTail
$ns duplex-link $n5 $n6 1Mb 1ms DropTail
$ns duplex-link $n6 $n7 1Mb 1ms DropTail
$ns duplex-link $n7 $n8 1Mb 1ms DropTail
$ns duplex-link $n8 $n0 1Mb 1ms DropTail
$ns duplex-link $n0 $n10 1Mb 1ms DropTail
$ns duplex-link $n9 $n1 1Mb 1ms DropTail

set bgp_agent0 [$n0 get-bgp-agent]
$bgp_agent0 bgp-id 10.0.0.1
$bgp_agent0 neighbor 10.0.1.1 remote-as 1
$bgp_agent0 neighbor 10.0.2.1 remote-as 2
$bgp_agent0 neighbor 10.0.3.1 remote-as 3
$bgp_agent0 neighbor 10.0.4.1 remote-as 4
$bgp_agent0 neighbor 10.0.8.1 remote-as 8
$bgp_agent0 neighbor 10.0.10.1 remote-as 10
$bgp_agent0 dampening 1 0 3000 750 900 1000 500 3600

set bgp_agent1 [$n1 get-bgp-agent]
$bgp_agent1 bgp-id 10.0.1.1
$bgp_agent1 neighbor 10.0.0.1 remote-as 0
$bgp_agent1 neighbor 10.0.3.1 remote-as 3
$bgp_agent1 neighbor 10.0.5.1 remote-as 5
$bgp_agent1 neighbor 10.0.9.1 remote-as 9
$bgp_agent1 dampening 1 0 3000 750 900 1000 500 3600

set bgp_agent2 [$n2 get-bgp-agent]
$bgp_agent2 bgp-id 10.0.2.1
```

```
$bgp_agent2 neighbor 10.0.0.1 remote-as 0
$bgp_agent2 neighbor 10.0.4.1 remote-as 4
$bgp_agent2 dampening 1 0 3000 750 900 1000 500 3600

set bgp_agent3 [$n3 get-bgp-agent]
$bgp_agent3 bgp-id 10.0.3.1
$bgp_agent3 neighbor 10.0.0.1 remote-as 0
$bgp_agent3 neighbor 10.0.1.1 remote-as 1
$bgp_agent3 neighbor 10.0.4.1 remote-as 4
$bgp_agent3 dampening 1 0 3000 750 900 1000 500 3600

set bgp_agent4 [$n4 get-bgp-agent]
$bgp_agent4 bgp-id 10.0.4.1
$bgp_agent4 neighbor 10.0.0.1 remote-as 0
$bgp_agent4 neighbor 10.0.2.1 remote-as 2
$bgp_agent4 neighbor 10.0.3.1 remote-as 3
$bgp_agent4 dampening 1 0 3000 750 900 1000 500 3600

set bgp_agent5 [$n5 get-bgp-agent]
$bgp_agent5 bgp-id 10.0.5.1
$bgp_agent5 neighbor 10.0.1.1 remote-as 1
$bgp_agent5 neighbor 10.0.6.1 remote-as 6
$bgp_agent5 dampening 1 0 3000 750 900 1000 500 3600

set bgp_agent6 [$n6 get-bgp-agent]
$bgp_agent6 bgp-id 10.0.6.1
$bgp_agent6 neighbor 10.0.5.1 remote-as 5
$bgp_agent6 neighbor 10.0.7.1 remote-as 7
$bgp_agent6 dampening 1 0 3000 750 900 1000 500 3600

set bgp_agent7 [$n7 get-bgp-agent]
$bgp_agent7 bgp-id 10.0.7.1
$bgp_agent7 neighbor 10.0.6.1 remote-as 6
$bgp_agent7 neighbor 10.0.8.1 remote-as 8
$bgp_agent7 dampening 1 0 3000 750 900 1000 500 3600

set bgp_agent8 [$n8 get-bgp-agent]
$bgp_agent8 bgp-id 10.0.8.1
$bgp_agent8 neighbor 10.0.0.1 remote-as 0
$bgp_agent8 neighbor 10.0.7.1 remote-as 7
$bgp_agent8 dampening 1 0 3000 750 900 1000 500 3600

set bgp_agent9 [$n9 get-bgp-agent]
$bgp_agent9 bgp-id 10.0.9.1
$bgp_agent9 neighbor 10.0.1.1 remote-as 1
$bgp_agent9 dampening 1 0 3000 750 900 1000 500 3600

set bgp_agent10 [$n10 get-bgp-agent]
$bgp_agent10 bgp-id 10.0.10.1
$bgp_agent10 neighbor 10.0.0.1 remote-as 0
$bgp_agent10 dampening 1 0 3000 750 900 1000 500 3600

$ns at 1.25 "puts \"\n time: 1.25 \n n10 (ip_addr 10.0.10.1) \
                       defines a network 10.1.2.0/24.\""
$ns at 1.25 "$bgp_agent10 network 10.1.2.0/24"

$ns at 31.0 "puts \"\n time: 31 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 31.0 "$bgp_agent0 show-routes"
$ns at 31.0 "$bgp_agent0 show-all"
$ns at 31.0 "$bgp_agent1 show-routes"
$ns at 31.0 "$bgp_agent1 show-all"
$ns at 31.0 "$bgp_agent2 show-routes"
$ns at 31.0 "$bgp_agent2 show-all"
$ns at 31.0 "$bgp_agent3 show-routes"
$ns at 31.0 "$bgp_agent3 show-all"
$ns at 31.0 "$bgp_agent4 show-routes"
$ns at 31.0 "$bgp_agent4 show-all"
$ns at 31.0 "$bgp_agent5 show-routes"
$ns at 31.0 "$bgp_agent5 show-all"
$ns at 31.0 "$bgp_agent6 show-routes"
$ns at 31.0 "$bgp_agent6 show-all"
$ns at 31.0 "$bgp_agent7 show-routes"
$ns at 31.0 "$bgp_agent7 show-all"
$ns at 31.0 "$bgp_agent8 show-routes"
$ns at 31.0 "$bgp_agent8 show-all"
$ns at 31.0 "$bgp_agent9 show-routes"
$ns at 31.0 "$bgp_agent9 show-all"
$ns at 31.0 "$bgp_agent10 show-routes"
$ns at 31.0 "$bgp_agent10 show-all"

$ns at 35.35 "puts \"\n time: 35.35 \n n1 (ip_addr 10.0.10.1) \
                       withdraws the network 10.1.2.0/24.\""
$ns at 35.35 "$bgp_agent10 no-network 10.1.2.0/24"

$ns at 38.0 "puts \"\n time: 38 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 38.0 "$bgp_agent0 show-routes"
$ns at 38.0 "$bgp_agent0 show-all"
$ns at 38.0 "$bgp_agent1 show-routes"
$ns at 38.0 "$bgp_agent1 show-all"
$ns at 38.0 "$bgp_agent2 show-routes"
$ns at 38.0 "$bgp_agent2 show-all"
$ns at 38.0 "$bgp_agent3 show-routes"
$ns at 38.0 "$bgp_agent3 show-all"
$ns at 38.0 "$bgp_agent4 show-routes"
$ns at 38.0 "$bgp_agent4 show-all"
$ns at 38.0 "$bgp_agent5 show-routes"
$ns at 38.0 "$bgp_agent5 show-all"
$ns at 38.0 "$bgp_agent6 show-routes"
$ns at 38.0 "$bgp_agent6 show-all"
$ns at 38.0 "$bgp_agent7 show-routes"
$ns at 38.0 "$bgp_agent7 show-all"
$ns at 38.0 "$bgp_agent8 show-routes"
$ns at 38.0 "$bgp_agent8 show-all"
$ns at 38.0 "$bgp_agent9 show-routes"
$ns at 38.0 "$bgp_agent9 show-all"
$ns at 38.0 "$bgp_agent10 show-routes"
$ns at 38.0 "$bgp_agent10 show-all"

$ns at 98.0 "$bgp_agent10 network 10.1.2.0/24"
$ns at 130.0 "$bgp_agent10 no-network 10.1.2.0/24"
$ns at 200.0 "$bgp_agent10 network 10.1.2.0/24"
```

```
$ns at 248.0 "puts \"\n time: 248 \
                    \n dump routing tables in all BGP agents: \n\""
$ns at 248.0 "$bgp_agent0 show-routes"
$ns at 248.0 "$bgp_agent0 show-all"
$ns at 248.0 "$bgp_agent0 show-damping"
$ns at 248.0 "$bgp_agent1 show-routes"
$ns at 248.0 "$bgp_agent1 show-all"
$ns at 248.0 "$bgp_agent1 show-damping"
$ns at 248.0 "$bgp_agent2 show-routes"
$ns at 248.0 "$bgp_agent2 show-all"
$ns at 248.0 "$bgp_agent2 show-damping"
$ns at 248.0 "$bgp_agent3 show-routes"
$ns at 248.0 "$bgp_agent3 show-all"
$ns at 248.0 "$bgp_agent3 show-damping"
$ns at 248.0 "$bgp_agent4 show-routes"
$ns at 248.0 "$bgp_agent4 show-all"
$ns at 248.0 "$bgp_agent4 show-damping"
$ns at 248.0 "$bgp_agent5 show-routes"
$ns at 248.0 "$bgp_agent5 show-all"
$ns at 248.0 "$bgp_agent5 show-damping"
$ns at 248.0 "$bgp_agent6 show-routes"
$ns at 248.0 "$bgp_agent6 show-all"
$ns at 248.0 "$bgp_agent6 show-damping"
$ns at 248.0 "$bgp_agent7 show-routes"
$ns at 248.0 "$bgp_agent7 show-all"
$ns at 248.0 "$bgp_agent7 show-damping"
$ns at 248.0 "$bgp_agent8 show-routes"
$ns at 248.0 "$bgp_agent8 show-all"
$ns at 248.0 "$bgp_agent8 show-damping"
$ns at 248.0 "$bgp_agent9 show-routes"
$ns at 248.0 "$bgp_agent9 show-all"
$ns at 248.0 "$bgp_agent9 show-damping"
$ns at 248.0 "$bgp_agent10 show-routes"
$ns at 248.0 "$bgp_agent10 show-all"
$ns at 248.0 "$bgp_agent10 show-damping"

$ns at 250.0 "finish"

proc finish {} {
      global ns
      puts "Simulation finished. "
      exit 0
}

puts "Simulation starts..."
$ns run
```

APPENDIX B FIGURES AND TABLES FOR OCCASIONAL FLAPS

This Appendix contains figures and tables that are supplementary to those included in Section 4. These figures and tables show the performance of three existing RFD algorithms (*original RFD*, *selective RFD*, and *RFD+*) for five BRITE-generated networks (100 – 500 nodes) in the case of occasional flaps. As in Section 4.1, the following aspects of BGP performance are examined: advertisement and withdrawal phases (B.1), effect of inter-arrival time between route updates (B.2), location of the origin router (B.3), impact of route suppression (B.4), and flaps identified by individual nodes (B.5). In each case, the results agree with those presented in Section 4.1.

B.1 Advertisement and Withdrawal Phases

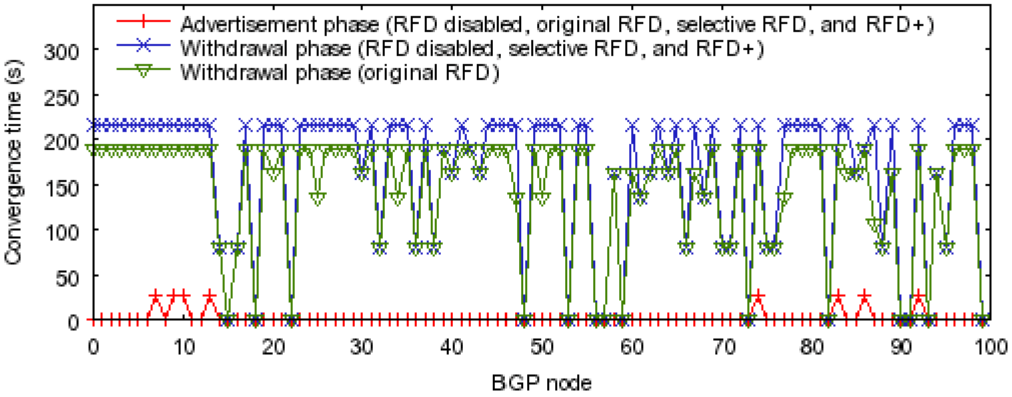

Figure B.1: Occasional flaps: convergence times for individual BGP nodes during the advertisement and withdrawal phases in a 100-node network.

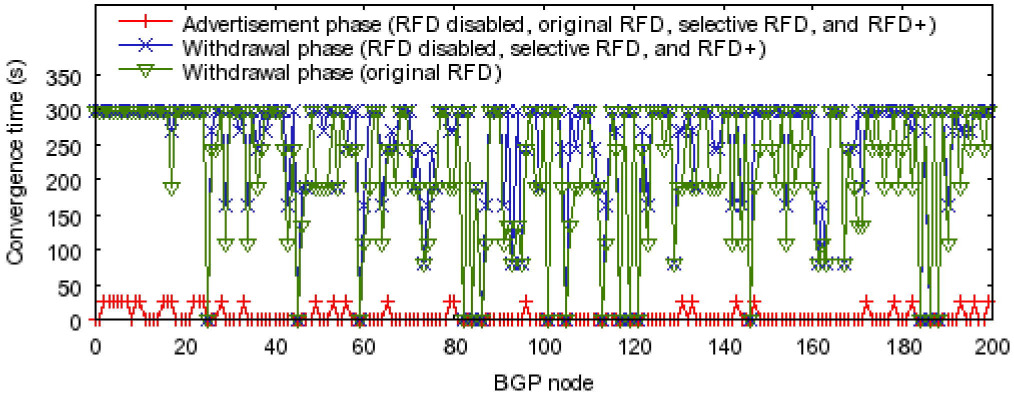

Figure B.2: **Occasional flaps: convergence times for individual BGP nodes during the advertisement and withdrawal phases in a 200-node network.**

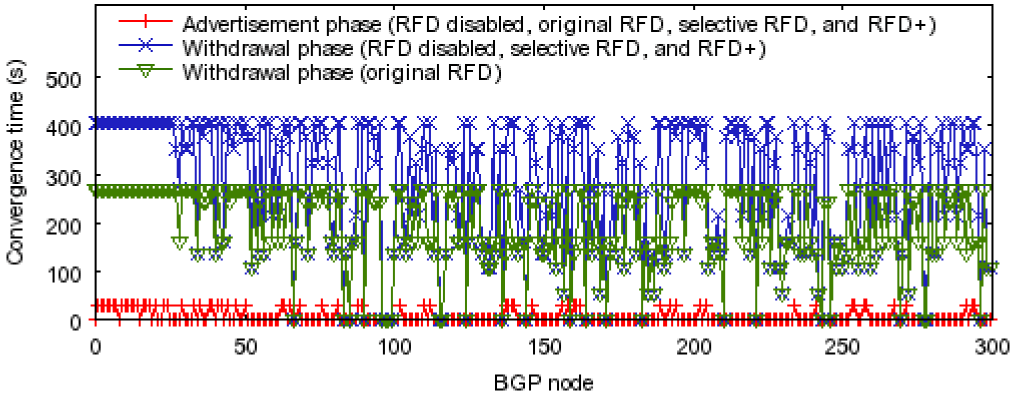

Figure B.3: **Occasional flaps: convergence times for individual BGP nodes during the advertisement and withdrawal phases in a 300-node network.**

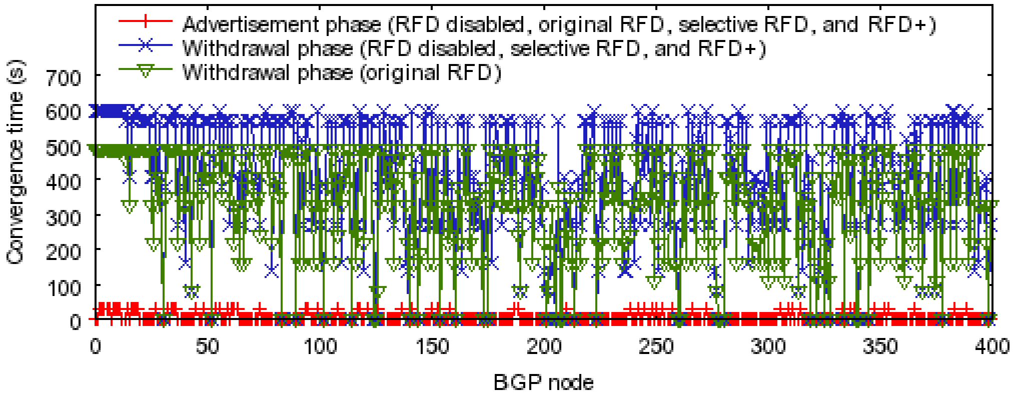

Figure B.4: **Occasional flaps: convergence times for the individual BGP nodes during advertisement and withdrawal phases in a 400-node network.**

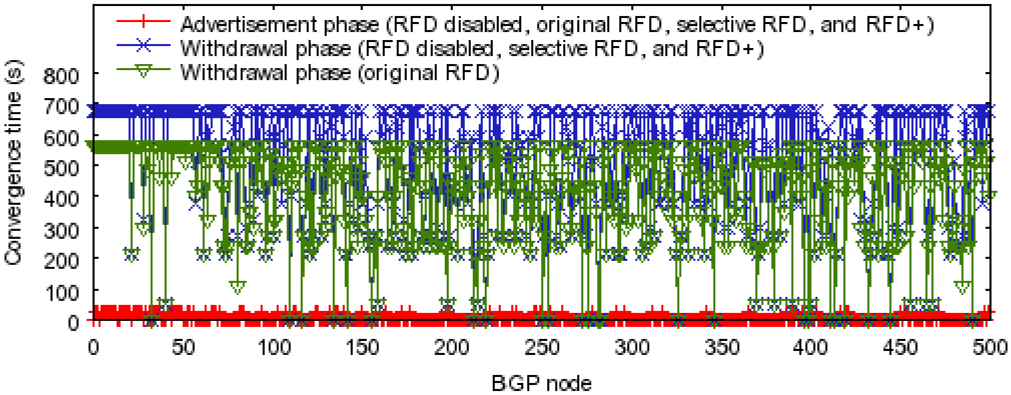

Figure B.5: Occasional flaps: convergence times for individual BGP nodes during the advertisement and withdrawal phases in a 500-node network.

Table B.1: Occasional flaps in *RFD disabled*: comparison between the advertisement and withdrawal phases.

Phase	Evaluation parameters	Network size (no. of nodes)				
		100	200	300	400	500
Advertisement phase	Convergence time (s)	27.017	27.017	27.017	27.017	27.018
	Average time (s)	2.177	4.470	5.954	4.469	5.468
	No. of updates	275	552	959	1169	1626
Withdrawal phase	Convergence time (s)	216.21	297.31	405.3	594.21	675.3
	Average time (s)	162.93	243.90	279.93	397.61	497.35
	No. of updates	2857	6952	12208	21006	33390

Table B.2: Occasional flaps in *original RFD*: comparison between the advertisement and withdrawal phases.

Phase	Evaluation parameters	Network size (no. of nodes)				
		100	200	300	400	500
Advertisement phase	Convergence time (s)	27.017	27.017	27.017	27.017	27.018
	Average time (s)	2.177	4.470	5.954	4.469	5.468
	No. of updates	275	552	959	1169	1626
Withdrawal phase	Convergence time (s)	189.21	297.31	270.31	486.21	567.21
	Average time (s)	141.59	218.41	193.79	323.70	410.56
	No. of updates	2450	6237	8695	16896	26892

Table B.3: Occasional flaps in *selective RFD*: comparison between the advertisement and withdrawal phases.

Phase	Evaluation parameters	Network size (no. of nodes)				
		100	200	300	400	500
Advertisement phase	Convergence time (s)	27.017	27.017	27.017	27.017	27.018
	Average time (s)	2.177	4.470	5.954	4.469	5.468
	No. of updates	275	552	959	1169	1626
Withdrawal phase	Convergence time (s)	216.21	297.31	405.3	594.21	675.3
	Average time (s)	162.93	243.90	279.93	397.61	497.35
	No. of updates	2857	6952	12208	21006	33390

Table B.4: Occasional flaps in *RFD+*: comparison between the advertisement and withdrawal phases.

Phase	Evaluation parameters	Network size (no. of nodes)				
		100	200	300	400	500
Advertisement phase	Convergence time (s)	27.017	27.017	27.017	27.017	27.018
	Average time (s)	2.177	4.470	5.954	4.469	5.468
	No. of updates	275	552	959	1169	1626
Withdrawal phase	Convergence time (s)	216.21	297.31	405.3	594.21	675.3
	Average time (s)	162.93	243.90	279.93	397.61	497.35
	No. of updates	2857	6952	12208	21006	33390

B.2 Effect of Inter-arrival Time between Route Updates

Table B.5: Occasional flaps: effect of inter-arrival time on network performance in a 100-node network.

Algorithm	Evaluation parameters	Inter-arrival time between updates (s)								
		10	20	30	50	100	300	500	800	1000
RFD disabled	Convergence time (s)	61.02	41.02	51.02	31.02	35.02	27.02	27.02	27.02	27.02
	No. of updates	859	930	1267	1267	1931	3407	3407	3407	3407
	No. of flaps	-	-	-	-	-	-	-	-	-
	No. of suppressions	-	-	-	-	-	-	-	-	-

Algorithm	Evaluation parameters	Inter-arrival time between updates (s)								
		10	20	30	50	100	300	500	800	1000
Original RFD	Convergence time (s)	61.02	41.02	51.02	31.02	35.01	0.02	0.02	0.02	0.02
	No. of updates	859	930	1267	1267	1919	2965	2971	2971	2971
	No. of flaps	476	495	701	701	1074	1502	1502	1502	1502
	No. of suppressions	0	0	3	3	187	199	175	175	175
Selective RFD	Convergence time (s)	61.02	41.02	51.02	31.02	35.02	27.02	27.02	27.02	27.02
	No. of updates	859	930	1267	1267	1931	3407	3407	3407	3407
	No. of flaps	259	276	299	299	289	334	334	334	334
	No. of suppressions	0	0	0	0	0	0	0	0	0
RFD+	Convergence time (s)	61.02	41.02	51.02	31.02	35.02	27.02	27.02	27.02	27.02
	No. of updates	859	930	1267	1267	1931	3407	3407	3407	3407
	No. of flaps	256	256	258	258	257	261	261	261	261
	No. of suppressions	0	0	0	0	0	0	0	0	0

Table B.6: Occasional flaps: effect of inter-arrival time on network performance in a 200-node network.

Algorithm	Evaluation parameters	Inter-arrival time between updates (s)								
		10	20	30	50	100	300	500	800	1000
RFD disabled	Convergence time (s)	61.02	41.02	48.02	8.02	35.02	51.02	27.02	27.02	27.02
	No. of updates	1832	1930	2434	2434	3725	8052	8056	8056	8056
	No. of flaps	-	-	-	-	-	-	-	-	-
	No. of suppressions	-	-	-	-	-	-	-	-	-
Original RFD	Convergence time (s)	61.02	41.02	48.02	8.02	35.02	24.12	0.02	0.02	0.02
	No. of updates	1832	1930	2434	2434	3678	7169	7202	7202	7202
	No. of flaps	961	1001	1430	1430	2229	3800	3829	3829	3829
	No. of suppressions	0	0	8	8	398	418	395	395	395
Selective RFD	Convergence time (s)	61.02	41.02	48.02	8.02	35.02	51.02	27.02	27.02	27.02
	No. of updates	1832	1930	2434	2434	3725	8051	8055	8056	8056
	No. of flaps	487	530	563	563	624	800	800	802	802
	No. of suppressions	0	0	0	0	3	6	5	3	3

Algorithm	Evaluation parameters	Inter-arrival time between updates (s)								
		10	20	30	50	100	300	500	800	1000
RFD+	Convergence time (s)	61.02	41.02	48.02	8.02	35.02	51.02	27.02	27.02	27.02
	No. of updates	1832	1930	2434	2434	3725	8052	8056	8056	8056
	No. of flaps	435	454	491	491	493	496	497	497	497
	No. of suppressions	0	0	0	0	0	0	0	0	0

Table B.7: Occasional flaps: effect of inter-arrival time on network performance in a 300-node network.

Algorithm	Evaluation parameters	Inter-arrival time between updates (s)		
		10	100	1000
RFD disabled	Convergence time (s)	61.02	8.21	27.02
	No. of updates	2794	5637	14126
	No. of flaps	-	-	-
	No. of suppressions	-	-	-
Original RFD	Convergence time (s)	61.02	35.01	0.02
	No. of updates	2794	5599	10371
	No. of flaps	1643	3712	6196
	No. of suppressions	0	650	658
Selective RFD	Convergence time (s)	61.02	8.21	27.02
	No. of updates	2794	5637	14126
	No. of flaps	800	903	1166
	No. of suppressions	0	0	6
RFD+	Convergence time (s)	61.02	8.21	27.02
	No. of updates	2794	5637	14126
	No. of flaps	724	796	813
	No. of suppressions	0	0	0

Table B.8: Occasional flaps: effect of inter-arrival time on network performance in a 400-node network.

Algorithm	Evaluation parameters	Inter-arrival time between updates (s)		
		10	100	1000
RFD disabled	Convergence time (s)	61.02	35.02	27.02
	No. of updates	3631	7318	23344
	No. of flaps	-	-	-
	No. of suppressions	-	-	-
Original RFD	Convergence time (s)	61.02	35.02	27.01
	No. of updates	3631	7274	18979
	No. of flaps	2052	4756	11181
	No. of suppressions	0	853	880
Selective RFD	Convergence time (s)	61.02	35.02	27.02
	No. of updates	3631	7318	23344
	No. of flaps	1001	1221	1590
	No. of suppressions	0	0	13
RFD+	Convergence time (s)	61.02	35.02	27.02
	No. of updates	3631	7318	23344
	No. of flaps	952	1019	1049
	No. of suppressions	0	0	0

Table B.9: Occasional flaps: effect of inter-arrival time on network performance in a 500-node network.

Algorithm	Evaluation parameters	Inter-arrival time between updates (s)		
		10	100	1000
RFD disabled	Convergence time (s)	61.02	35.02	27.02
	No. of updates	4743	9677	36642
	No. of flaps	-	-	-
	No. of suppressions	-	-	-
Original RFD	Convergence time (s)	61.02	35.02	0.02
	No. of updates	4743	9577	29662
	No. of flaps	2838	6418	17877
	No. of suppressions	0	1150	1194
Selective RFD	Convergence time (s)	61.02	35.02	27.02
	No. of updates	4743	9677	36642
	No. of flaps	1353	1602	1880
	No. of suppressions	0	0	6
RFD+	Convergence time (s)	61.02	35.02	27.02
	No. of updates	4743	9677	36642
	No. of flaps	1256	1362	1386
	No. of suppressions	0	0	0

B.3 Location of the Origin Router

Table B.10: Occasional flaps in *RFD disabled*: effect of the origin router's location.

Phase	Location	Evaluation parameters	Network size (no. of nodes)				
			100	200	300	400	500
Advertisement phase	Connected to core	Convergence time (s)	27.017	27.017	27.017	27.017	27.018
		No. of updates	275	552	959	1169	1626
	Connected to edge	Convergence time (s)	27.019	54.019	27.019	81.018	54.02
		No. of updates	275	670	1011	1978	2939
Withdrawal phase	Connected to core	Convergence time (s)	216.21	297.31	405.3	594.21	675.3
		No. of updates	2857	6952	12208	21006	33390
	Connected to edge	Convergence time (s)	216.21	351.3	486.21	594.41	756.31
		No. of updates	2857	8122	15189	24527	36820

Table B.11: Occasional flaps in *original RFD*: effect of the origin router's location.

Phase	Location	Evaluation parameters	Network size (no. of nodes)				
			100	200	300	400	500
Advertisement phase	Connected to core	Convergence time (s)	27.017	27.017	27.017	27.017	27.018
		No. of updates	275	552	959	1169	1626
	Connected to edge	Convergence time (s)	27.019	54.019	27.019	81.018	54.02
		No. of updates	275	670	1011	1978	2939
Withdrawal phase	Connected to core	Convergence time (s)	189.21	297.31	270.31	486.21	567.21
		No. of updates	2450	6237	8695	16896	26892
	Connected to edge	Convergence time (s)	189.21	270.3	405.2	486.31	594.31
		No. of updates	2450	5531	12321	19535	28488

Table B.12: Occasional flaps in *selective RFD*: effect of the origin router's location.

Phase	Location	Evaluation parameters	Network size (no. of nodes)				
			100	200	300	400	500
Advertisement phase	Connected to core	Convergence time (s)	27.017	27.017	27.017	27.017	27.018
		No. of updates	275	552	959	1169	1626
	Connected to edge	Convergence time (s)	27.019	54.019	27.019	81.018	54.02
		No. of updates	275	670	1011	1978	2939
Withdrawal phase	Connected to core	Convergence time (s)	216.21	297.31	405.3	594.21	675.3
		No. of updates	2857	6952	12208	21006	33390
	Connected to edge	Convergence time (s)	216.21	351.3	486.21	594.41	756.31
		No. of updates	2857	8122	15189	24527	36820

Table B.13: Occasional flaps in *RFD+*: effect of the origin router's location.

Phase	Location	Evaluation parameters	Network size (no. of nodes)				
			100	200	300	400	500
Advertisement phase	Connected to core	Convergence time (s)	27.017	27.017	27.017	27.017	27.018
		No. of updates	275	552	959	1169	1626
	Connected to edge	Convergence time (s)	27.019	54.019	27.019	81.018	54.02
		No. of updates	275	670	1011	1978	2939
Withdrawal phase	Connected to core	Convergence time (s)	216.21	297.31	405.3	594.21	675.3
		No. of updates	2857	6952	12208	21006	33390
	Connected to edge	Convergence time (s)	216.21	351.3	486.21	594.41	756.31
		No. of updates	2857	8122	15189	24527	36820

B.4 Impact of Route Suppression

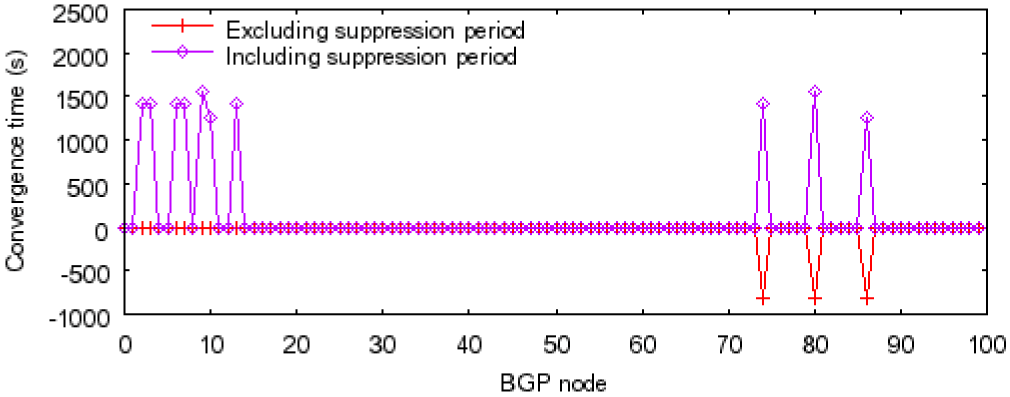

Figure B.6: Occasional flaps in *original RFD*: impact of route suppression in a network with 100 nodes.

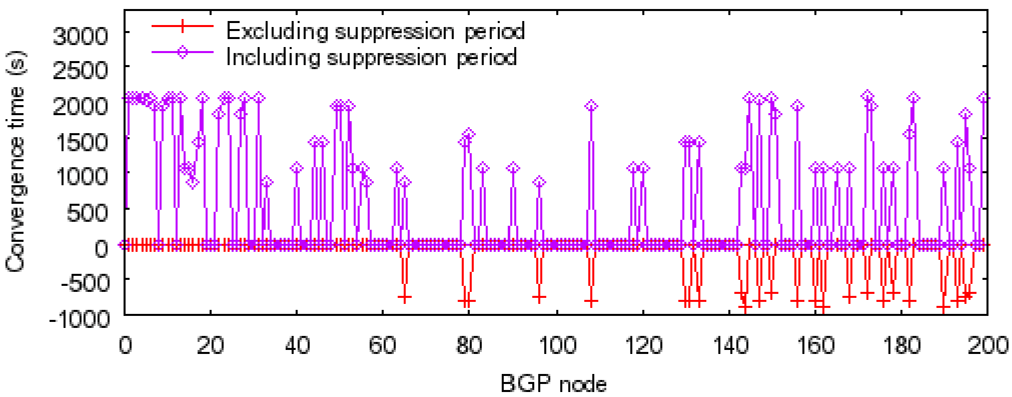

Figure B.7: Occasional flaps in *original RFD*: impact of route suppression in a network with 200 nodes.

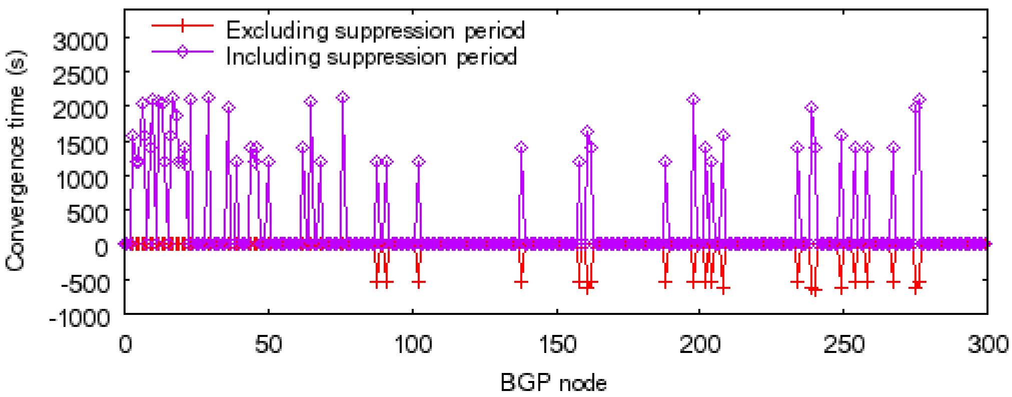

Figure B.8: Occasional flaps in *original RFD*: impact of route suppression in a network with 300 nodes.

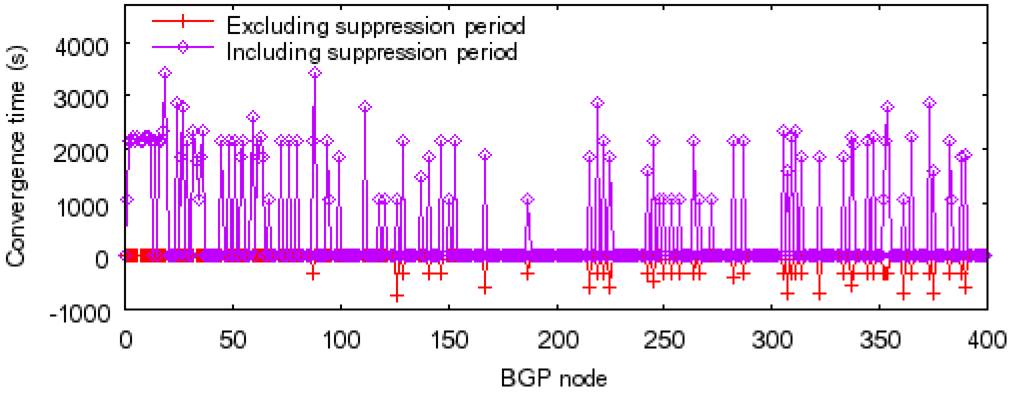

Figure B.9: Occasional flaps in *original RFD*: impact of route suppression in a network with 400 nodes.

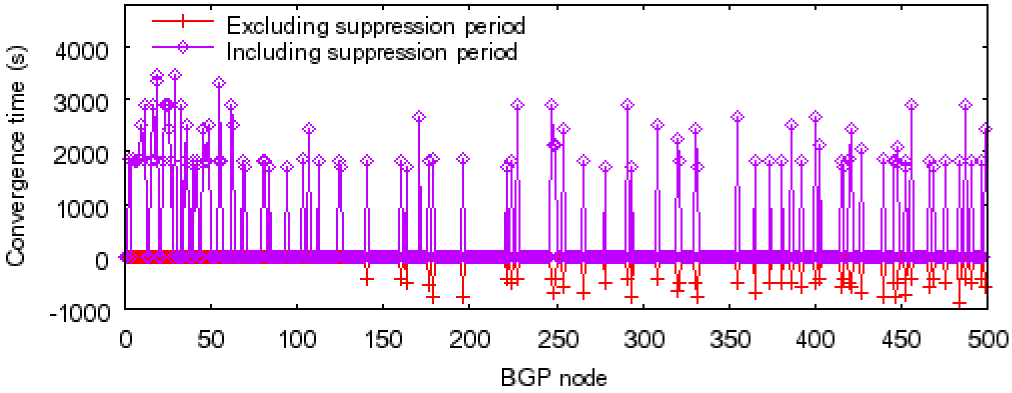

Figure B.10: Occasional flaps in *original RFD*: impact of route suppression in a network with 500 nodes.

B.5 Flaps Identified by Individual Nodes

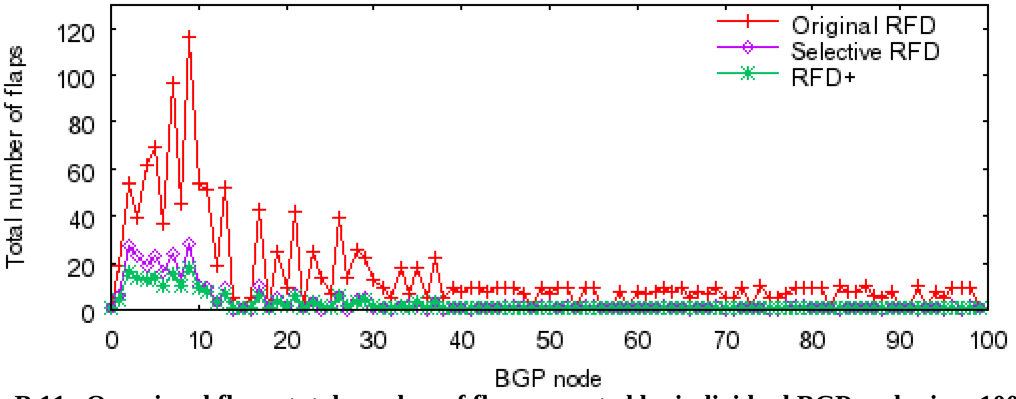

Figure B.11: Occasional flaps: total number of flaps reported by individual BGP nodes in a 100-node network.

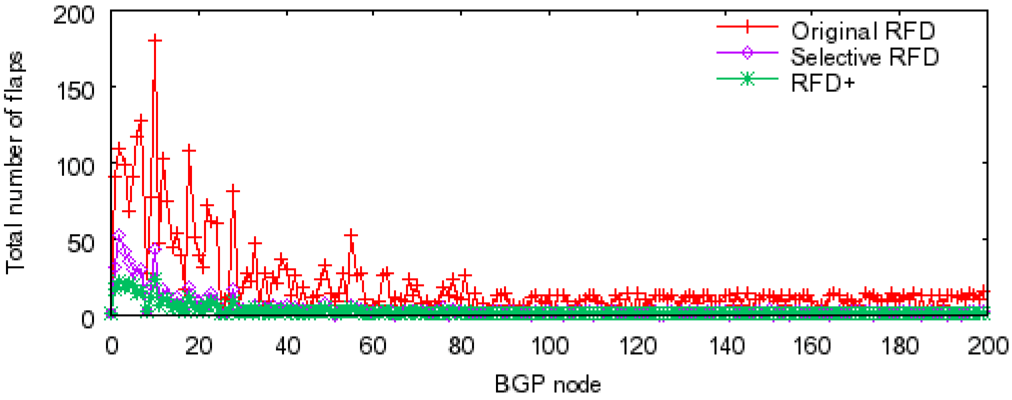

Figure B.12: Occasional flaps: total number of flaps reported by individual BGP nodes in a 200-node network.

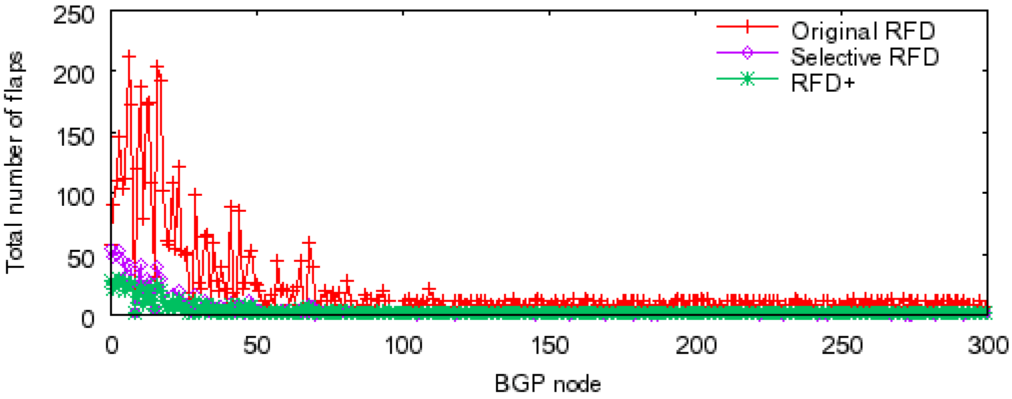

Figure B.13: Occasional flaps: total number of flaps reported by individual BGP nodes in a 300-node network.

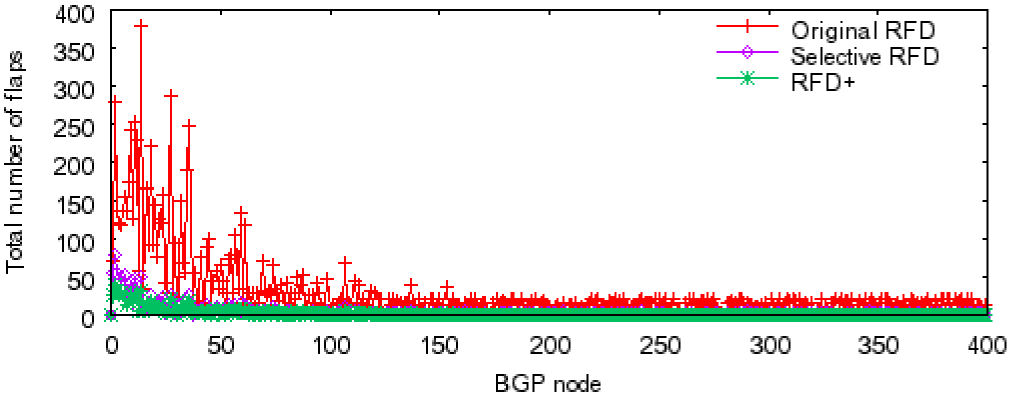

Figure B.14: Occasional flaps: total number of flaps reported by individual BGP nodes in a 400-node network.

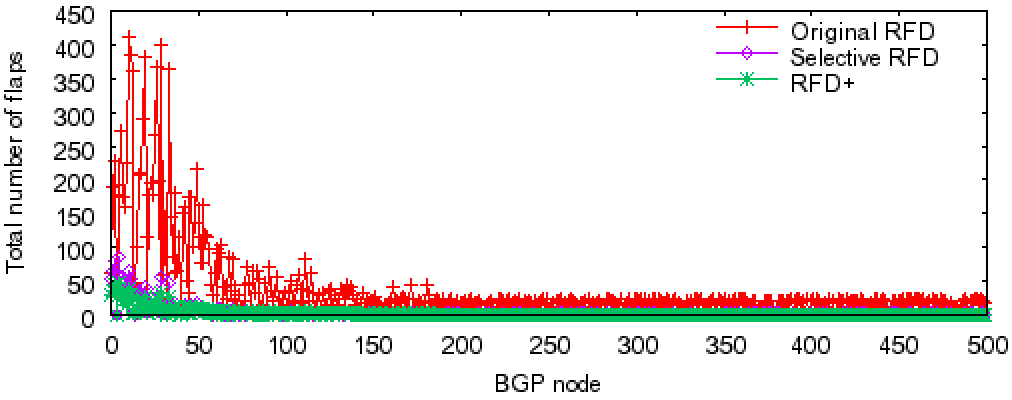

Figure B.15: Occasional flaps: total number of flaps reported by individual BGP nodes in a 500-node network.

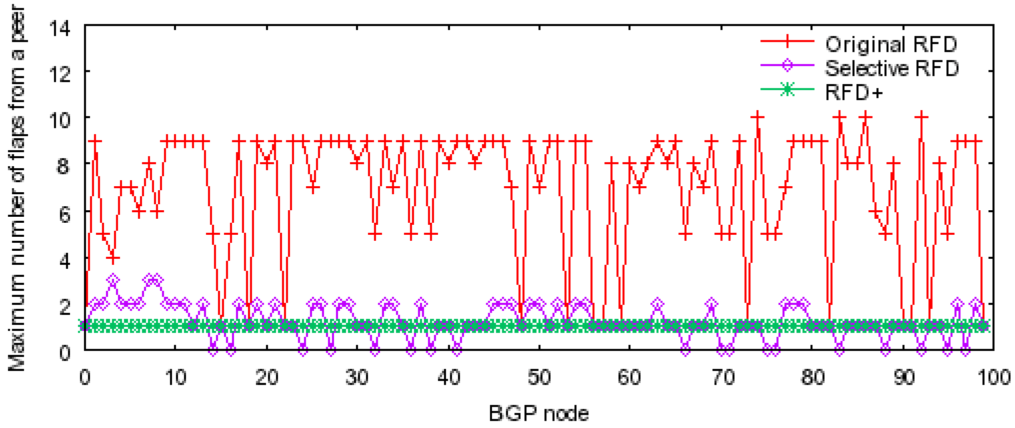

Figure B.16: Occasional flaps: maximum number of flaps reported by each BGP node for one of its peers in a 100-node network.

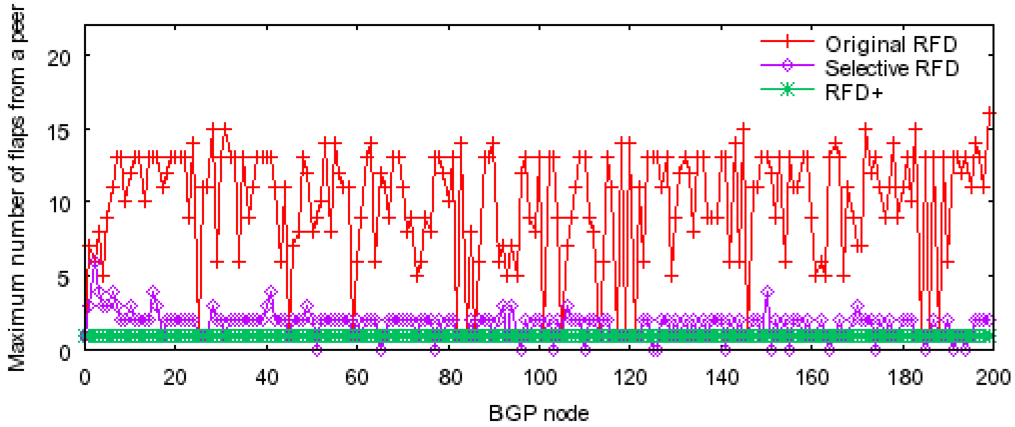

Figure B.17: Occasional flaps: maximum number of flaps reported by each BGP node for one of its peers in a 200-node network.

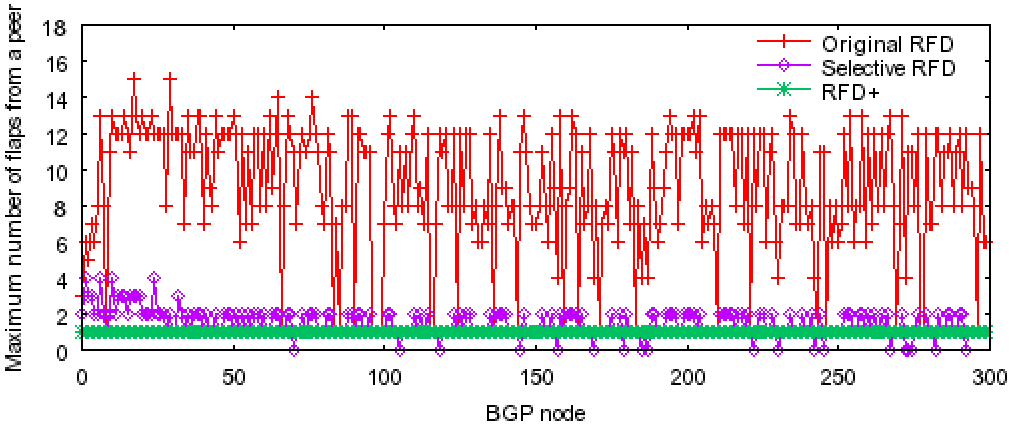

Figure B.18: Occasional flaps: maximum number of flaps reported by each BGP node for one of its peers in a 300-node network.

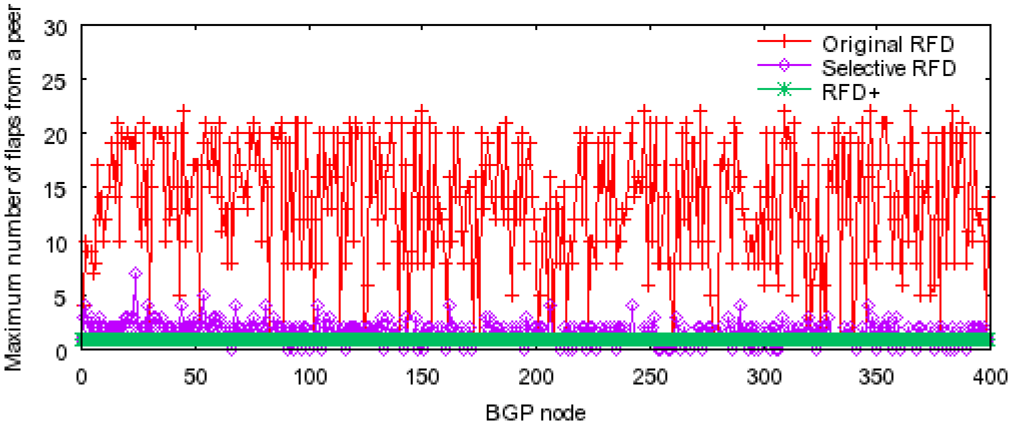

Figure B.19: Occasional flaps: maximum number of flaps reported by each BGP node for one of its peers in a 400-node network.

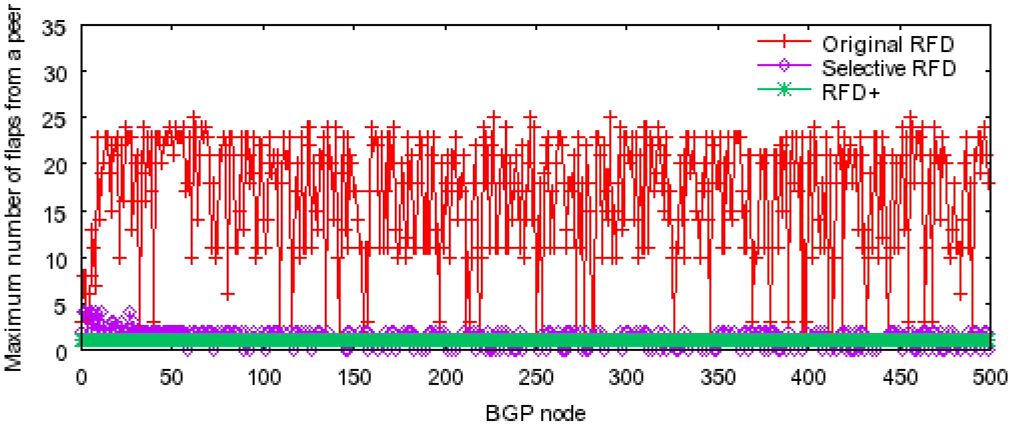

Figure B.20: Occasional flaps: maximum number of flaps reported by each BGP node for one of its peers in a 500-node network.

Table B.14: Occasional flaps: comparison of the number of reported flaps between three RFD algorithms.

Algorithm	Evaluation parameters	Network size (no. of nodes)				
		100	200	300	400	500
Original RFD	Total number of flaps for a single node (maximum)	116 (20 peers)	180 (27 peers)	212 (36 peers)	379 (46 peers)	412 (57 peers)
	Total number of flaps for a single node (minimum)	1	1	1	1	1
	Maximum number of flaps associated with a single peer of a node	10	16	15	22	25
Selective RFD	Total number of flaps for a single node (maximum)	28 (20 peers)	53 (38 peers)	54 (38 peers)	80 (71 peers)	86 (70 peers)
	total number of flaps for a single node (minimum)	0	0	0	0	0
	Maximum number of flaps associated with a single peer of a node	3	6	4	5	4
RFD+	Total number of flaps for a single node (maximum)	18 (20 peers)	24 (27 peers)	29 (42 peers)	40 (71 peers)	46 (70 peers)
	total number of flaps for a single node (minimum)	1	1	1	1	1
	Maximum number of flaps associated with a single peer of a node	1	1	1	1	1

APPENDIX C FIGURES AND TABLES FOR PERSISTENT FLAPS

This Appendix contains figures and tables that are supplementary to those included in Section 5. These figures and tables show the performance of three existing RFD algorithms (*original RFD*, *selective RFD*, and *RFD+*) for all simulated network topologies in the case of persistent flaps. We examine the convergence times of individual BGP nodes and the maximum number of flaps reported by each BGP node for one of its peers. In each case, the results agree with those presented in Section 5.1.

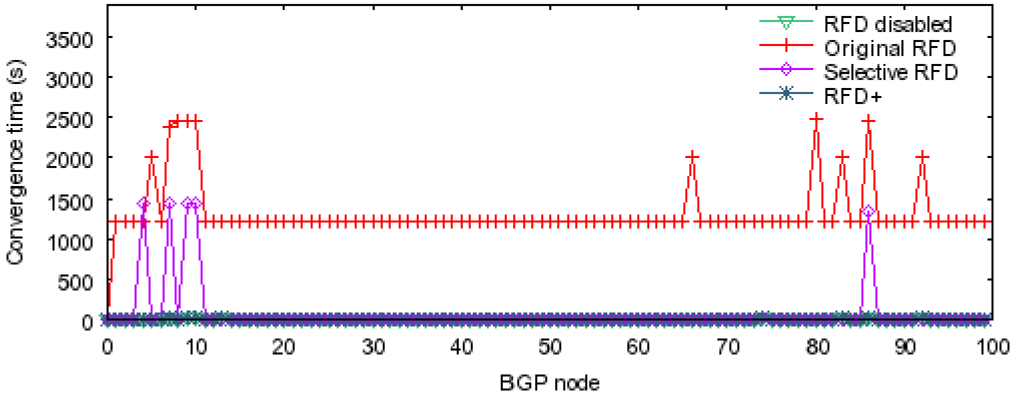

Figure C.1: Persistent flaps: convergence times for individual BGP nodes in a 100-node network.

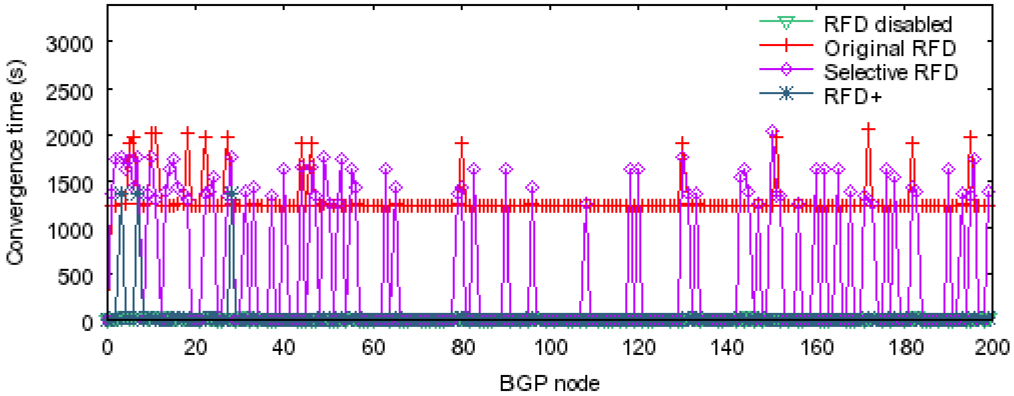

Figure C.2: Persistent flaps: convergence times for individual BGP nodes in a 200-node network.

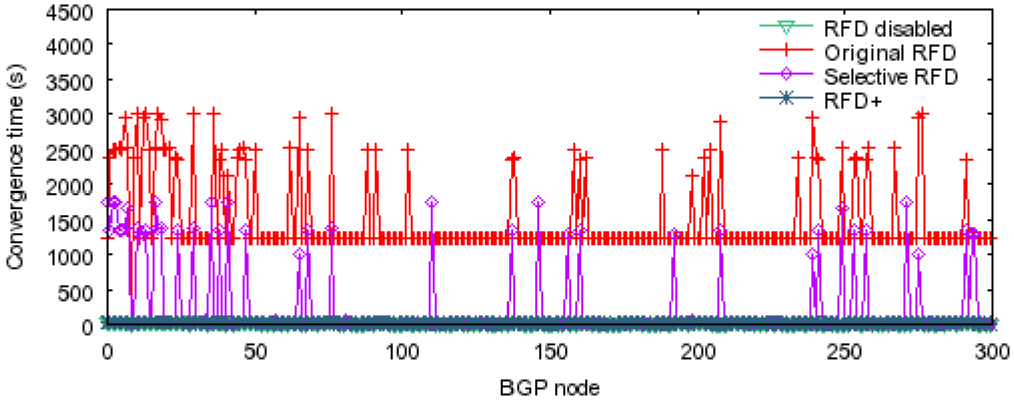

Figure C.3: Persistent flaps: convergence times for individual BGP nodes in a 300-node network.

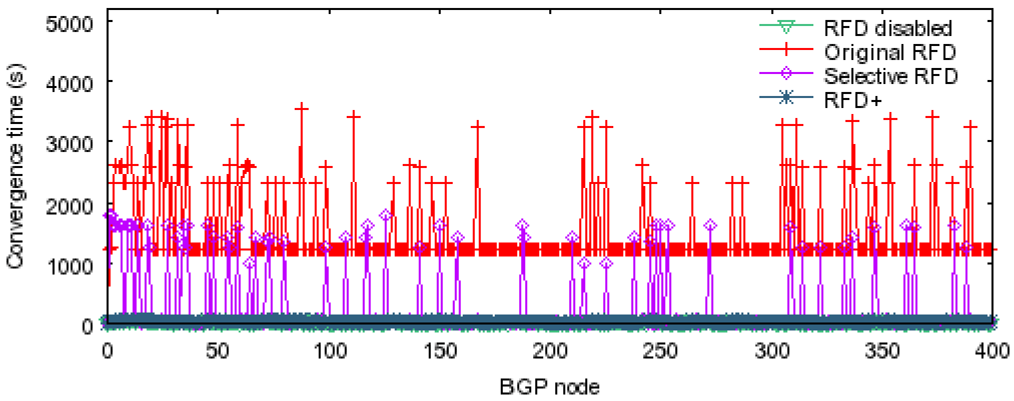

Figure C.4: Persistent flaps: convergence times for individual BGP nodes in a 400-node network.

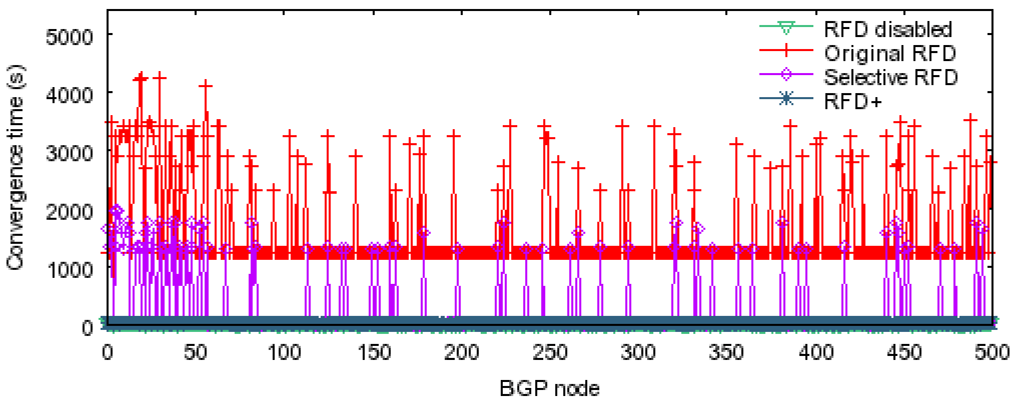

Figure C.5: Persistent flaps: convergence times for individual BGP nodes in a 500-node network.

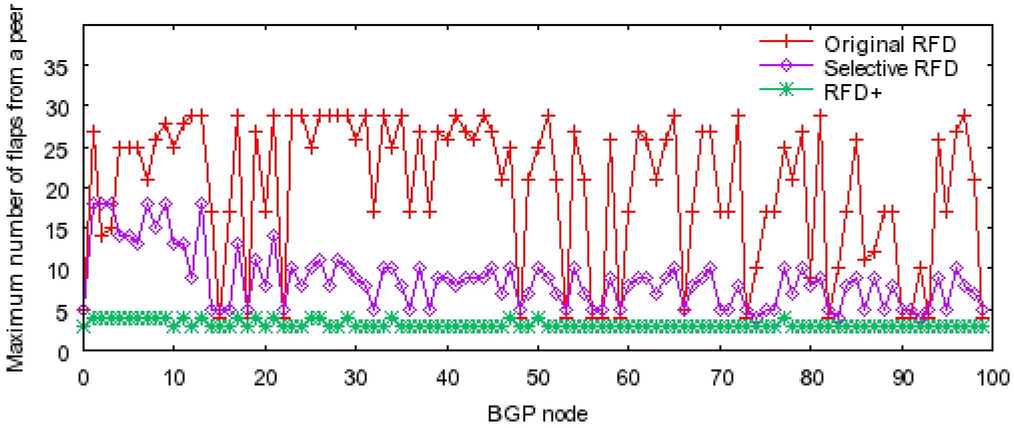

Figure C.6: **Persistent flaps: maximum number of flaps reported by each BGP node for one of its peers in a 100-node network.**

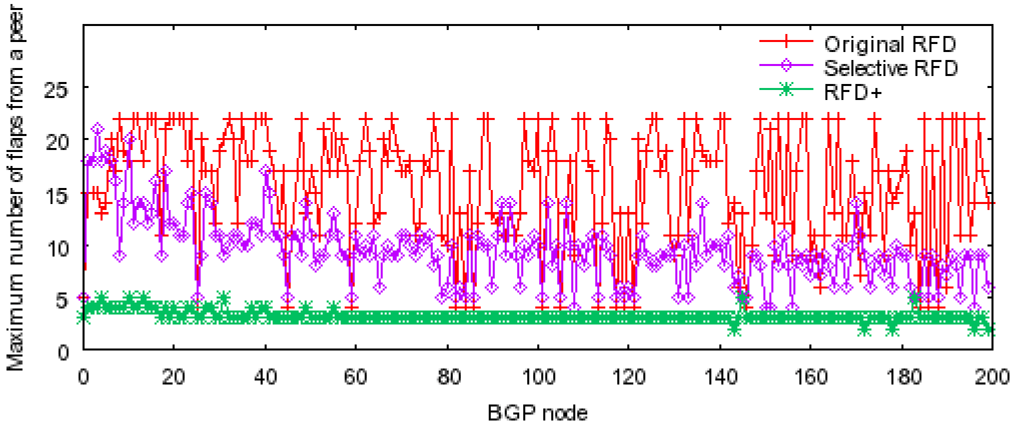

Figure C.7: Persistent flaps: maximum number of flaps reported by each BGP node for one of its peers in a 200-node network.

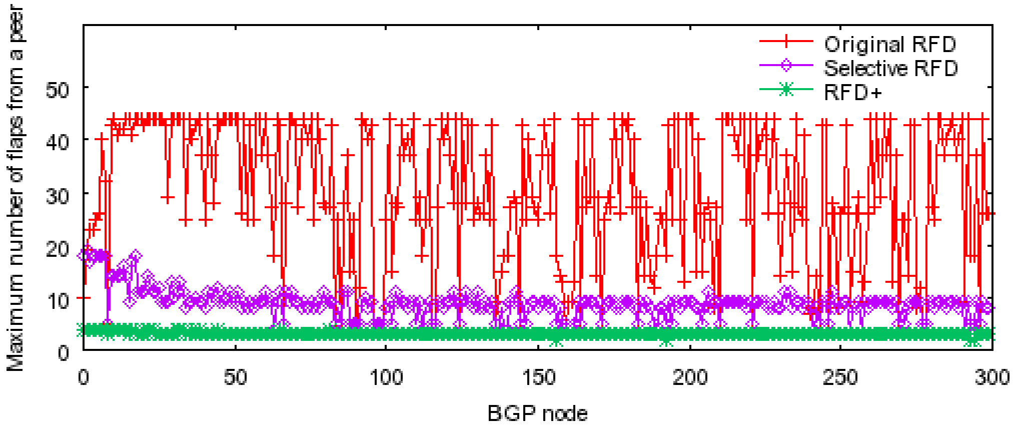

Figure C.8: Persistent flaps: maximum number of flaps reported by each BGP node for one of its peers in a 300-node network.

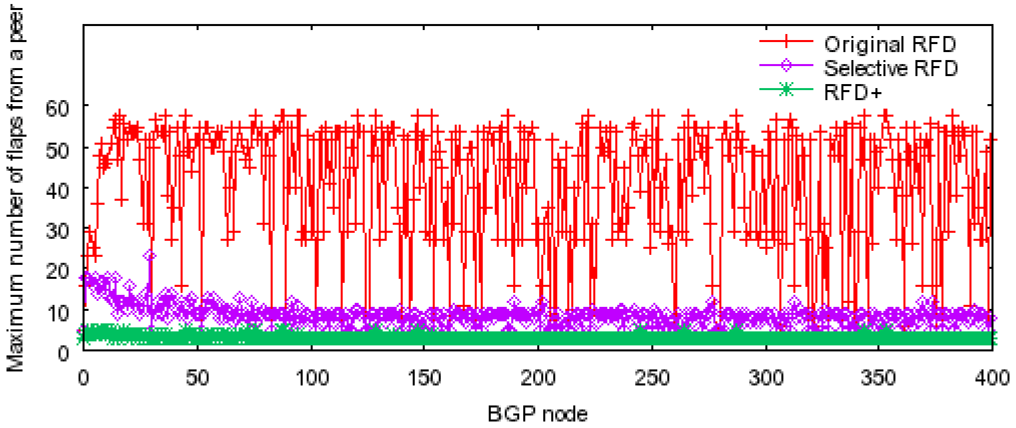

Figure C.9: **Persistent flaps: maximum number of flaps reported by each BGP node for one of its peers in a 400-node network.**

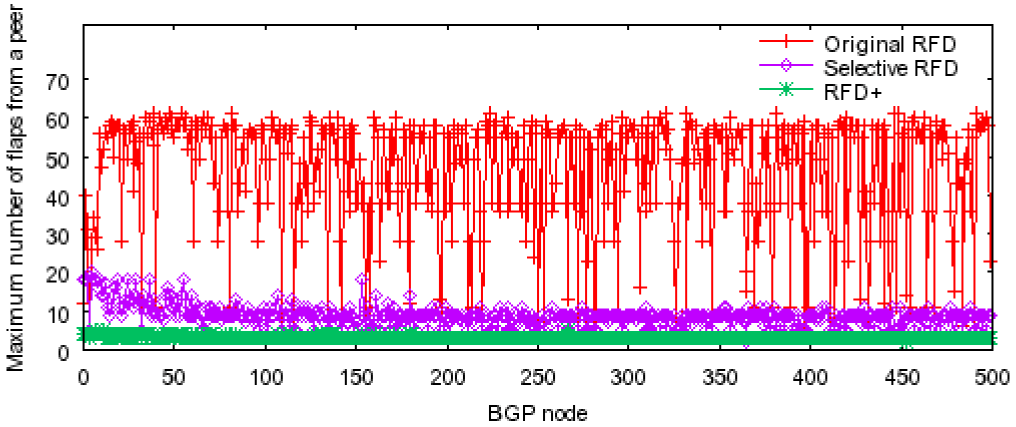

Figure C.10: **Persistent flaps: maximum number of flaps reported by each BGP node for one of its peers in a 500-node network.**

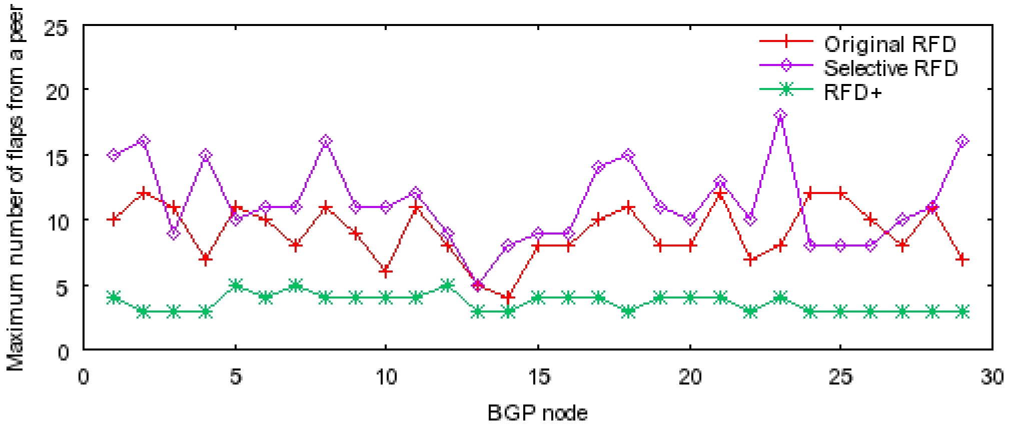

Figure C.11: **Persistent flaps: maximum number of flaps reported by each BGP node for one of its peers in a 29-node network.**

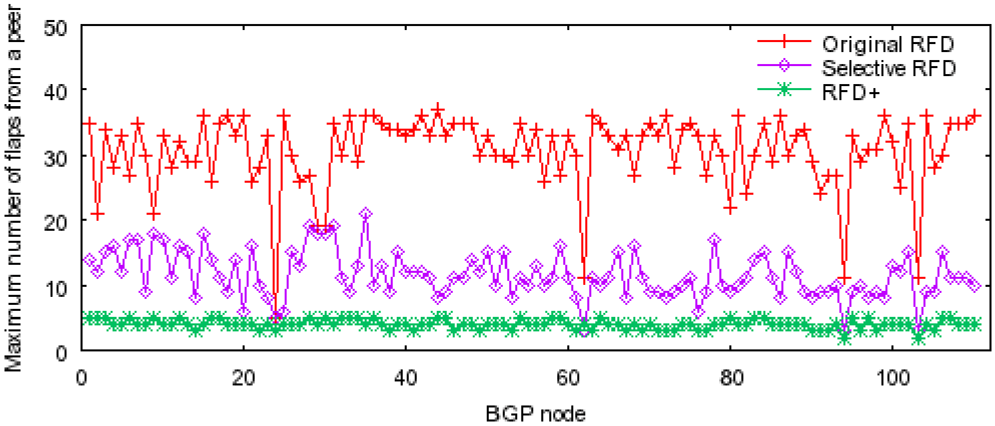

Figure C.12: Persistent flaps: maximum number of flaps reported by each BGP node for one of its peers in a 110-node network.

Table C.1: Persistent flaps in BRITE-generated topologies: comparison of the number of flaps between three existing RFD algorithms.

Algorithm	Evaluation parameters	Network size (no. of updates)				
		100	200	300	400	500
Original RFD	Total number of flaps for a single node (maximum)	324 (20 peers)	275 (27 peers)	601 (36 peers)	1234 (46 peers)	1378 (45 peers)
	Total number of flaps for a single node (minimum)	4	4	4	4	4
	Maximum number of flaps associated with a single peer of a node	29	22	44	58	61
Selective RFD	Total number of flaps for a single node (maximum)	243 (21 peers)	352 (38 peers)	411 (38 peers)	653 (71 peers)	580 (70 peers)
	Total number of flaps for a single node (minimum)	4	4	3	5	5
	Maximum number of flaps associated with a single peer of a node	18	21	19	23	20
RFD+	Total number of flaps for a single node (maximum)	64 (20 peers)	89 (38 peers)	107 (42 peers)	186 (71 peers)	157 (70 peers)
	Total number of flaps for a single node (minimum)	3	2	2	3	2
	Maximum number of flaps associated with a single peer of a node	4	5	4	5	5

Table C.2: Persistent flaps in topologies built from routing tables: comparison of the number of flaps between three existing RFD algorithms.

Algorithm	Evaluation parameters	Network size (no. of nodes)	
		29	110
Original RFD	Total number of flaps for a single node (maximum)	41 (8 peers)	403 (22 peers)
	Total number of flaps for a single node (minimum)	5	5
	Maximum number of flaps associated with a single peer of a node	12	37
Selective RFD	Total number of flaps for a single node (maximum)	50 (7 peers)	256 (20 peers)
	Total number of flaps for a single node (minimum)	5	5
	Maximum number of flaps associated with a single peer of a node	18	21
RFD+	Total number of flaps for a single node (maximum)	27 (8 peers)	76 (20 peers)
	Total number of flaps for a single node (minimum)	3	3
	Maximum number of flaps associated with a single peer of a node	5	5

BIBLIOGRAPHY

[1] Y. Rekhter and T. Li, “A border gateway protocol 4 (BGP-4),” *RFC 1771*, Mar. 1995.

[2] Y. Rekhter and P. Gross, “Application of the border gateway protocol in the Internet,” *RFC 1772*, Mar. 1995.

[3] A. Colton, *Cisco IOS for IP Routing*. London: Rocket Science Press, 2001.

[4] S. Halabi and D. McPherson, *Internet Routing Architectures*. Indianapolis, IN: Cisco Press, 2000.

[5] J. Doyle and J. D. Carroll, *Routing TCP/IP (Vol. II).* Indianapolis, IN: Cisco Press, 2001.

[6] C. Huitema, *Routing in the Internet*. Upper Saddle River, NJ: Prentice Hall, 2000.

[7] Cisco IOS IP Configuration Guide. (2006, January). [Online]. Available: http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_configuration_guide_book09186a0080087fa9.html.

[8] Cisco Documentation: Using the Border Gateway Protocol for Interdomain Routing. (2006, January). [Online]. Available: http://www.cisco.com/univercd/cc/td/doc/cisintwk/ics/icsbgp4.htm.

[9] C. Labovitz, G. Malan, and F. Jahanian, “Origins of Internet routing instability,” in *Proc. INFOCOM*, New York, NY, Mar. 1999, pp. 218–226.

[10] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, “Delayed Internet routing convergence,” *IEEE/ACM Transactions on Networking*, vol. 9, no. 3, pp. 293–306, June 2001.

[11] C. Labovitz, R. Wattenhofer, S. Venkatachary, and A. Ahuja, “The impact of Internet policy and topology on delayed routing convergence,” in *Proc. INFOCOM*, Anchorage, AK, Apr. 2001, pp. 537–546.

[12] R. Govindan and A. Reddy, “An analysis of Internet inter-domain topology and route stability,” in *Proc. INFOCOM 1997*, Kobe, Japan, Apr. 1997, pp. 850–857.

[13] K. Varadhan, R. Govindan, and D. Estrin, “Persistent route oscillations in inter-domain routing,” *Computer Networks*, vol. 32, no. 1, pp. 1–16, Jan. 2000.

[14] C. Villamizar, R. Chandra, and R. Govindan, “BGP route flap damping,” *RFC 2439*, Nov. 1998.

[15] T. Griffin, “What is the sound of one route flapping?,” *Network Modelling and Simulation Summer Workshop*, Dartmouth College, NH, July 2002.

[16] Z. Mao, R. Govindan, G. Varghese, and R. Katz, “Route flap damping exacerbates Internet routing convergence,” in *Proc. SIGCOMM*, Pittsburgh, PA, Aug. 2002, pp. 221–233.

[17] Z. Duan, J. Chandrashekar, J. Krasky, K. Xu, and Z.-L. Zhang, “Damping BGP route flaps,” in *Proc. IPCCC*, Phoenix, AZ, Apr. 2004, pp. 131–138.

[18] W. Shen and Lj. Trajkovic, “BGP route flap damping algorithms,” in *Proc. SPECTS’05*, Philadelphia, PA, July 2005, pp. 488–495.

[19] ns-BGP 2.0. (2006, January). [Online]. Available: http://www.ensc.sfu.ca/~ljilja/cnl/projects/BGP.

[20] T. D. Feng, R. Ballantyne, and Lj. Trajković, “Implementation of BGP in a network simulator,” in *Proc. ATS’04*, Arlington, VA, Apr. 2004, pp. 149–154.

[21] T. D. Feng, “Implementation of BGP in a network simulator,” M.Sc. thesis, Simon Fraser University, BC, Canada, Apr. 2004.

[22] The network simulator ns-2. (2006, January). [Online]. Available: http://www.isi.edu/nsnam/ns.

[23] SSFNet. (2006, January). [Online]. Available: http://www.ssfnet.org/homePage.html.

[24] RFD-AMRAI BGP. (2006, January). [Online]. Available: http://www.ensc.sfu.ca/~ljilja/cnl/projects/RFD-AMRAI.

[25] T. G. Griffin and B. J. Premore, “An experimental analysis of BGP convergence time,” in *Proc. ICNP 2001*, Riverside, CA, Nov. 2001, pp. 53–61.

[26] T. Griffin and G. Wilfong, “An analysis of BGP convergence properties,” in *Proc. SIGCOMM 1999*, Cambridge, MA, Sept. 1999, pp. 277–288.

[27] A. Feldmann, H. Kong, O. Maennel, and A. Tudor, “Measuring BGP pass-through times,” in *Proc. PAM*, Antibes Juan-les-Pins, France, Apr. 2004, pp. 267–277.

[28] N. Lasković and Lj. Trajković, “BGP with an adaptive minimal route advertisement interval,” to appear in *Proc. IPCCC 2006*, Phoenix, AZ, April, 2006.

[29] R. Bush, T. Griffin, and Z. Morley Mao, "Route flap damping: harmful?" *NANOG 26*, Eugene, OR, Oct. 2002. (2006, January). [Online]. Available: http://www.nanog.org/mtg-0210/ppt/flap.pdf.

[30] The ns Manual. (2006, January). [Online]. Available: http://www.isi.edu/nsnam/ns/ns-documentation.html.

[31] BRITE. (2006, January). [Online]. Available: http://www.cs.bu.edu/brite.

[32] T. Bu and D. Towsley, “On distinguishing between Internet power law topology generators,” in *Proc. INFOCOM*, New York, NY, June 2002, pp. 638–647.

[33] Multi-AS topologies from routing tables. (2006, January). [Online]. Available: http://www.ssfnet.org/Exchange/gallery/asgraph.

[34] C. Panigl, J. Schmitz, P. Smith, and C. Vistoli, “RIPE routing-WG recommendations for coordinated route-flap damping parameters,” Oct. 2001. (2006, January). [Online]. Available: http://www.ripe.net/ripe/docs/ripe-229.html.

[35] GnuPlot. (2006, January). [Online]. Available: http://www.gnuplot.info.

[36] Route Views Project. (2006, January). [Online]. Available: http://www.routeviews.org.

[37] RIS Statistics Report. (2006, January). [Online]. Available: http://www.ris.ripe.net/weekly-report/reports.

[38] S. Ilnicki, and A. Tudor, “Observations on redundant BGP traffic and flaps from the RIPE RIS collectors,” *RIPE 43 Meeting*, Rhodes, Greece, Sept. 2002. (2006, January). [Online]. Available: http://www.ripe.net/ripe/meetings/ripe-43/presentations/ripe43-routing-redundant.

[39] Route Views Data Archives. (2006, January). [Online]. Available: http://archive.routeviews.org.

[40] Z. Mao, R. Bush, T. Griffin, and M. Roughan, “BGP beacons,” in *Proc. Internet Measurement Conference 2003,* Miami Beach, FL, Oct. 2003, pp. 1–14.

[41] BGP Beacon Info. (2006, January). [Online]. Available: http://psg.com/~zmao/BGPBeacon.html.

[42] K. Sriram, private communication.