Timer Interaction in Route Flap Damping

Beichuan Zhang
bzhang@cs.ucla.edu
UCLA

Dan Pei
peidan@cs.ucla.edu
UCLA

Daniel Massey
massey@cs.colostate.edu
Colorado State University

Lixia Zhang
lixia@cs.ucla.edu
UCLA

Abstract

Route Flap Damping *is a mechanism generally used in network routing protocols. Its goal is to limit the global impact of unstable routes by temporarily suppressing routes with rapid changes over short time periods. Although route damping is a clearly defined and simple procedure at each router, its effect in a large network setting is not well understood. We show that the current damping design leads to the intended behavior only under persistent route flapping. When the number of flaps is small, the global routing dynamics deviates significantly from the expected behavior with a longer convergence delay. Previous work observed that a single route flap can falsely trigger route suppression due to path exploration. However our simulations show that this false suppression only accounts for 30% of the convergence delay after a single route flap. Our study reveals previously unknown interactions between reuse timers at different routers. Route suppression and reuse at different routers are triggered at different times and thus affect the number of updates received by other routers. In turn, this impacts other routers' damping behavior. We propose to use* ***Root Cause Notification*** *to eliminate both false suppression and undesirable timer interaction.*

1 Introduction

It remains a challenge to design a responsive and efficient routing protocol for large scale networks. In this paper we examine a specific problem caused by unexpected interactions among multiple nodes in a large network. Since dynamic routing protocols adapt to topological changes, a single unstable link can potentially cause a large number of updates being propagated throughout the entire network, consuming router CPU cycles and link bandwidth [8]. To limit the global impact of individual unstable routes, *Route Flap Damping* [14] was added to the Border Gateway Protocol (BGP) [12] several years ago. It is commonly believed that damping has played an essential role in putting the global Internet routing update overhead under control [3].

The goal of BGP damping is to allow updates of stable routes to pass through but block updates generated by unstable routes. Briefly, route flap damping works as follows. A router associates a penalty value with each destination (i.e., an IP prefix) advertised by a neighbor router. A route *flaps* whenever the neighbor router changes its route to the destination. When it happens, the penalty value is increased. In the absence of route changes, the penalty value decays over time. When the penalty value exceeds a predefined *cut-off* threshold, further updates from the same neighbor for the same destination will no longer be propagated. That is, the route is *suppressed*. When the penalty value drops below a predefined *reuse* threshold, the router will start propagating updates for that destination again, i.e., the route is *reused*. Throughout this paper we will use the word *damping* as an abbreviation for "route flap damping" to refer to the whole mechanism, and the word *suppression* to refer to the specific action of stopping propagating updates.

Despite its simple rules of operation at each router, the overall effect of damping is not fully understood. A recent study [9] showed that, after a *single* route flap, path exploration can falsely trigger route suppression and prolong the convergence time. Yet our simulations show that this false suppression alone can only account for 30% of the convergence delay after a single route flap, and cannot explain the damping behavior after two or more route flaps. We will show that the current BGP damping mechanism achieves the intended behavior only under *persistent* route flapping. When the number of route flaps is small, the global routing dynamics deviates significantly from the intended behavior. Because the current damping implementation counts all received updates in calculating the penalty value, and because route suppression and reuse at different routers happen at different times, false damping can be triggered not only by path exploration, but also by the updates due to route reuse at neighbor routers. We propose to add *Root Cause Notification* (RCN) [11] to routing updates, in order to eliminate both false suppression and undesirable timer interactions.

The remainder of the paper is organized as follows. Section 2 describes BGP damping mechanism and previous work. Section 3 analyzes the intended damping behavior.

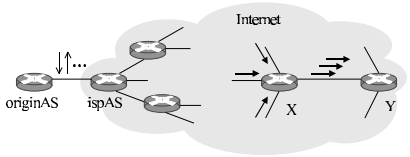


Figure 1. Example

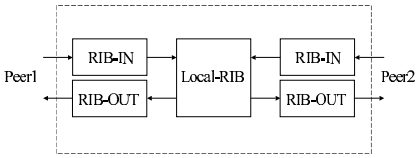


Figure 2. A router's RIBs

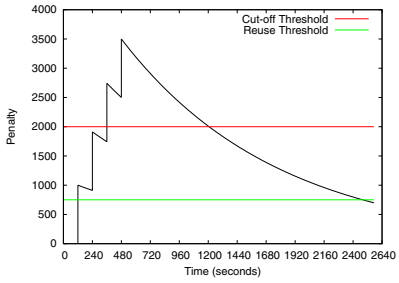


Figure 3. Damping Penalty

Section 4 analyzes the actual damping behavior in detail, including how timer interactions shape routing dynamics during damping. Section 5 presents simulation results. Section 6 proposes the use of RCN to facilitate damping. Section 7 discusses the implication of routing policies on damping, and Section 8 concludes the paper.

2 Route Flap Damping and Previous Work

Figure 1 shows a general network scenario that will be used throughout our analysis and simulations in this paper. A router in a customer network, the *originAS*, is connected to a router in its provider network, the *ispAS*. When the link $[originAS, ispAS]$ comes up, the router in ispAS will announce to the rest of the network the route to originAS; when the link goes down, the ispAS router will withdraw the route.

Generally speaking, each BGP router peers with a number of neighboring routers and exchanges routing updates. A router stores the routes received from each peer in the corresponding RIB-IN table (Figure 2). For each destination prefix, the router picks the best route among all the RIB-INs and stores this best route in the Local-RIB table. Depending on the routing policy, the router may announce all or part of its best routes to its peers. It stores the routes to be announced to each peer in the corresponding RIB-OUT table.

Damping associates a penalty value with each entry in a RIB-IN. That is, there is a penalty value associated with each peer and destination prefix pair. Whenever a new update message is received, the corresponding RIB-IN entry is updated and so is its penalty value. Different types of updates are assigned different penalty increments. If the penalty exceeds the *cut-off threshold*, the RIB-IN entry will no longer be used in selecting the best route. Note that during this route suppression, new routing updates for the same entry may continue to arrive, and if so the penalty value will continue to increase accordingly. Because a suppressed route does not enter Local-RIB, none of the new changes will be propagated any further.

Damping Parameters	Cisco	Juniper
Withdrawal Penalty (P_W)	1000	1000
Re-announcement Penalty (P_A)	0	1000
Attributes Change Penalty	500	500
Cut-off Threshold (P_{cut})	2000	3000
Half Life (minute) (H)	15	15
Reuse Threshold (P_{reuse})	750	750
Max Hold-down Time (minute)	60	60

Table 1. Default Damping Parameters

When the penalty value is greater than zero, it decays exponentially over time. More formally, if the penalty is $p(t_0)$ at time t_0 and becomes $p(t)$ at time t, then

$$p(t) = p(t_0)\, e^{-\lambda(t-t_0)} \tag{1}$$

where λ is often configured by the *half-life* $H = \ln 2/\lambda$. A suppressed route will be reused when the penalty drops below the *reuse threshold*. This is often implemented by setting a *reuse timer* based on the current penalty value and reusing the route when the reuse timer expires.

Table 1 lists the default parameter settings from two major router vendors, and Figure 3 shows an example of the penalty value (with Cisco default parameters) changes over time in response to a few route flaps.

Damping was introduced into Internet inter-domain routing in mid 1990s, and has been widely supported in commercial routers. RFC 2439 [14] documents its design rationale, algorithm, and implementation strategy. RFC 3221 [3] states that damping is widely deployed and helps stabilize the routing infrastructure, but it is also well known that different implementations use inconsistent parameters, and damping is not universally deployed everywhere.

The ***intended* effect of damping** is to allow occasional routing changes to propagate without delay, while suppress persistently changing routes until they become stable. However, as early as in 1998, Panigl [10] observed that a single route withdrawal followed by a re-announcement in Europe

triggered route suppression in North America. The cause of this behavior was not explained until 2002, when Mao *et al.* [9] showed that path exploration was the reason.

BGP path exploration was first reported by Labovitz *et al.* [6][7]. For example, in Figure 1, assume that X can reach originAS via three peers. When link $[originAS, ispAS]$ fails and ispAS sends a withdrawal to the rest of the network, X will receive the withdrawal from one of its peers first. Not knowing link $[originAS, ispAS]$ has failed, X will switch to another peer to reach originAS, thus it "explores" alternate paths. Every time X changes its route, it will send an update to Y. Only after receiving withdrawals from all its peers, will X finally send its own withdrawal to Y. This path exploration happens at every router in the network with alternative paths, and can amount to a large number of updates. Depending on the timing of these updates, Y can receive multiple updates on link $[X, Y]$ even though link $[originAS, ispAS]$ only flaps once.

[9] is the first work to point out the interplay between path exploration and damping. It shows that path exploration can amplify one single flap into many updates which falsely trigger suppression somewhere in the network. This unexpected interplay highlights the complexity introduced by the scale of a large network: one cannot easily predict the overall network behavior even if he knows exactly how each individual node works.

However, false suppression caused by path exploration alone cannot fully explain the observed long convergence delay. In [9], the simulation results show that convergence delay can be as long as one hour. In section 5.2, we will explain why it is unlikely to reach such a high penalty value by just path exploration. Moreover, [9] did not examine how damping would behave in response to more than one flap.

In this paper, we will give a detailed analysis of the damping process under one or more flaps, and show that it is the reuse timer interaction among multiple routers that stretches the convergence delay to be much longer than what path exploration alone could do. We will also show how to enhance the damping mechanism with RCN to prevent the undesirable behavior due to path exploration and reuse timer interaction.

3 The Intended Behavior of BGP Damping

Before analyzing its actual behavior in a network, we first quantify damping's intended behavior. We are interested in how damping affects routing dynamics in response to one or more route flaps. To quantify the effect, we use two metrics: *convergence time* and *message count*. Convergence time is defined as the time from when the originAS stops flapping (i.e., sends its final route announcement) to when the last update message is observed in the network. The message count is the total number of updates observed in the network starting from the first flap.

Convergence time can be calculated given the flapping intervals and damping parameters. For occasional flaps of link $[originAS, ispAS]$, route suppression should not be triggered, and the convergence time is the normal BGP convergence time, usually between seconds and a few minutes [6]. When link $[originAS, ispAS]$ flaps persistently, the excessive routing updates will increase the penalty value at ispAS and cause ispAS to suppress its route to originAS. After the flapping stops, ispAS will wait for the penalty value p to drop below the reuse threshold P_{reuse} before re-announcing the route. The announcement will trigger a BGP T_{up} event (i.e., a previously unreachable destination becomes reachable), which takes time t_{up} for the network to converge. Let r denote the time it takes for the penalty value to drop below the reuse threshold, then the total convergence time should be:

$$t = r + t_{up} \simeq r = \frac{1}{\lambda} \ln \frac{p}{P_{reuse}}$$

From Table 1, we can see that with Cisco default setting, r is at least 20 minutes and therefore $r \gg t_{up}$. The actual value of r depends on the penalty value p, the reuse threshold, and the half-life. To calculate p, let $w(i)$ be the time between the i^{th} flap and the $(i-1)^{th}$ flap, $f(i)$ be the penalty increment caused by the i^{th} flap, $i = 1, 2, \ldots, k-1, k$, and $w(1) = 0$. Right after the k^{th} flap, the penalty value $p(k)$ is

$$\begin{aligned} p(k) &= p(k-1) * e^{-\lambda w(k)} + f(k) \\ &= \sum_{i=1}^{k-1} [f(i) * e^{-\lambda \sum_{j=i+1}^{k} w(j)}] + f(k) \end{aligned}$$

Later in the paper, Figure 8 shows the calculation results of damping's intended convergence delay under a varying number of route flaps.

A precise message count generally cannot be obtained analytically, since it depends on the network topology and timing of updates. Nevertheless, the general trend can be predicted. As the number of flaps increases, the number of updates also increases since each new flap triggers some updates in the network. After a certain number of flaps, however, the message count is expected to be almost constant, since new flaps are suppressed by ispAS and no update is propagated beyond ispAS.

Damping reduces the number of updates by suppressing routing updates but it also increases convergence time. Our analysis suggests that ispAS can largely control the trade-off by setting appropriate penalty increments, cut-off threshold, and reuse threshold. The configuration can be tuned so that a small number of flaps does not trigger any damping delay, while a large number of flaps is suppressed, keeping the overall updates injected into the network at a

reasonable level. Therefore, the overall intended behavior in a network relies only on how the unstable link flaps and how the incident routers set their damping parameters, regardless of the rest of the network.

4 Damping Behavior in Distributed Systems

The previous section describes damping's intended behavior based on the rules applied to each individual router. However, the overall behavior of a network cannot be directly derived by examining individual routers separately. As we will show in this section, the network damping behavior is largely driven by previously unknown reuse timer interactions among different routers.

4.1 Stages of Damping Behavior

Our simulation studies show that, when an unstable destination exists and all the routers in a network perform BGP damping, the whole network goes through different states during damping. We will first give definitions to these states, then explain them in more detail, and discuss two types of reuse timer interactions.

- ***Charging***: It starts with the first flap of the route to the unstable destination. During the charging period, routing updates are exchanged among routers and each update increases (charges) the router's damping penalty. This charging ends when there is no update in transit or waiting to be sent in the whole network.
- ***Suppression***: After charging, if there is at least one router whose best route is unavailable due to suppression, the network enters suppression state, which ends when a reuse timer expires *and* triggers a new routing update.
- ***Releasing***: This period follows the suppression period and lasts until all the routing updates have been delivered.
- ***Converged***: After releasing, the network enters converged state, where every route in each router's Local-RIB is the best route from all its RIB-IN entries. Note that some RIB-IN entries might still be suppressed, but they are not the best route and thus their unavailability makes no impact to Local-RIB.

Figure 4 illustrates the transitions between different states.[1] Some routing flaps make the network move from the converged state to charging state, during which updates are propagated in the network and each update increases the penalty value at the receiving router, until eventually either the flapping stops or the flapping routes are suppressed. [9] showed that path exploration can amplify a single flap during the charging period and falsely trigger route suppression. In the rest of this section, we will analyze other states and show that reuse timer interaction plays a major role in these states.

[1] In the real Internet, due to its large scale, different parts of the network may be in different state and these four states may not be clearly separated.

4.2 Secondary Charging Effect

After charging ends, there is no update in flight or queued for transmission. However, some routes may be suppressed by some routers. In other words, some routes in RIB-IN cannot be used in Local-RIB because their penalties are over the threshold. This can occur in both the converged state and the suppressed state. To understand the difference between these two states, one must determine whether the reuse timer will be silent or noisy when it expires.

Figure 5 shows an example of a ***silent*** reuse timer. Router A has received two routes, R_B and R_C, from neighbors B and C respectively. R_B is the best path and is currently installed in Local-RIB, while R_C is suppressed and cannot be considered as a candidate for use in Local-RIB. When the reuse timer for R_C expires, R_C will become available and A will re-run its path selection algorithm. However in this case, R_C is irrelevant and R_B remains as the best path. We say this reuse timer is *silent* since its expiration will have no effect on Local-RIB and will not trigger any update by A. The network is in a converged state if there is no reuse timer at all or every reuse timer is silent.

Figure 6 shows an example of a ***noisy*** reuse timer. Again router A has received R_B and R_C from B and C respectively. But in this case, R_B is currently suppressed and cannot be considered as a candidate for use in Local-RIB. When its reuse timer expires, R_B becomes available and A will select it as the new best path. A will update its Local-RIB and RIB-OUT, and announces this change to its neighbors. In turn this new message may cause A's neighbors to update their routes. The network is in the suppression state if there is no pending update, *and* at least one router has a noisy reuse timer waiting to expire.

When a noisy reuse timer expires, the network moves from the suppression state to the releasing state, during which messages triggered by noisy route reuse can charge remaining reuse timers. For example, consider nodes X and Y in Figure 1, and assume Y has suppressed link (X, Y). If a noisy reuse timer expires at X, it will trigger an update sent to Y. Although this update was not directly caused by route flapping, Y will follow the damping rule and increase its penalty value, thus Y's reuse timer is charged again. We call this type of interaction between X's reuse timer and Y's reuse timer the ***Secondary Charging effect***. Combined with

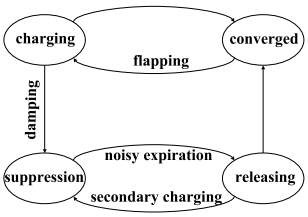


Figure 4. Four-state of a damping process in a distributed system

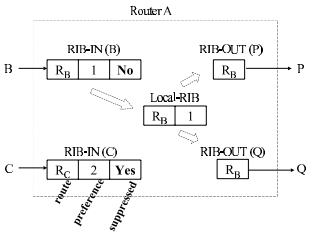


Figure 5. Silent Reuse

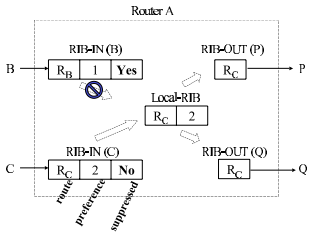


Figure 6. Noisy Reuse

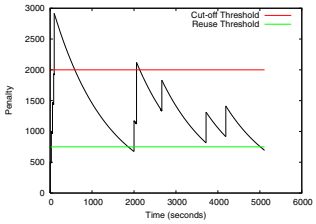


Figure 7. Damping Penalty

path exploration, secondary charging will not only lengthen existing reuse timers, but can also lead to new route suppressions sometimes. This drives the network to a new suppression period even though no new route flap has occurred. The network can converge only when all noisy reuse timers have expired.

Figure 7 shows an example of simulated route penalty over time after a single route flap. In this case, the router computing the penalty is not adjacent to the flapping link; more precisely it is 7 hops away from $originAS$. The charging period happens within the first 100 seconds, during which path exploration amplifies one flap into several updates and triggers route suppression, as described in [9]. With path exploration alone, the network would converge around 2000^{th} second when the route is reused. However, due to secondary charging, the penalty value is pushed up over the cut-off threshold again. Before the route is eventually reused after the 5000^{th} second, secondary charging pushes the penalty up three more times. In this case, secondary charging accounts for more than 60% of the total convergence delay! We will discuss details of the simulations in Section 5.

4.3 Muffling Effect

For a single or a small number of route flaps, ispAS does not suppress or delay any update. After a number of flaps, however, route suppression will be triggered at ispAS, and further flaps will be blocked from entering the network. Since the link $(originAS, ispAS)$ is suppressed, ispAS has no route to reach originAS. As a result, ispAS sends a route withdrawal to all its peers, which is then propagated throughout the network. Note that when a router receives a withdrawal message, it removes the route and increases the penalty value for that route. When a remote router's reuse timer expires, it will find no route to the originAS, thus cannot trigger any update. Any reuse timer that expires before ispAS reuses link $(originAS, ispAS)$ will be silent. We call this effect the ***Muffling effect***. The muffling effect is removed after ispAS reuses its route to the originAS and sends an announcement to the network.

4.4 Overall Damping Behavior

The above discussion shows that there are two types of reuse timer interaction: secondary charging prolongs convergence time, while muffling by ispAS' reuse timer reduces secondary charging by making potentially noisy timer expirations silent. These two types of timer interactions compete with each other, and the net result depends on the number of flaps sent by originAS.

Let RT_h be ispAS' reuse timer, and RT_{net} be the last noisy reuse timer in the rest of the network. Initially RT_h is zero as route suppression is not triggered at ispAS. But once it is triggered, any further flaps from originAS will increase RT_h only and have no effect on RT_{net} at all. As the number of flaps increases, a critical point (N_h) is reached when

$$RT_h > RT_{net}$$

That is, when the number of flaps is greater than a certain number N_h, RT_h will outlast all noisy reuse timers in the network, making the muffling effect dominant. When RT_h expires, it is the only reuse timer in the entire network, and there will be no secondary charging at all. The convergence time will be totally determined by when RT_h expires, which brings the convergence time in line with the intended behavior, as we described in Section 3. The overall results can be summarized as follows:

- After a small number of route flaps, due to path exploration and secondary charging, a network with damping can have longer convergence time than the intended behavior.
- When the number of flaps is greater than a certain number, due to muffling effect, a system with damping follows the intended behavior.

5 Simulation Results

Predicting the actual damping behavior in a network is difficult. It depends on the degree of path exploration, timing of updates, and order of reuse timer expirations. There-

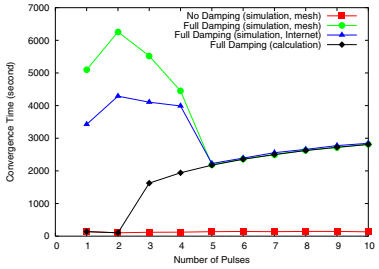


Figure 8. Convergence Time

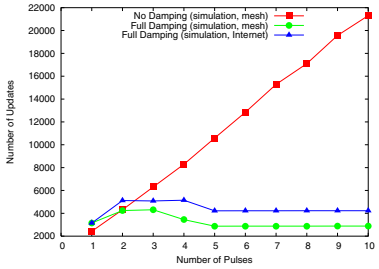


Figure 9. Message Count

fore we resort to simulations to verify our analysis and further illustrate the timer interactions.

5.1 Simulation Methodology

We conducted BGP simulations using SSFNet [13]. Two types of network topologies are used: mesh topologies and Internet-derived topologies. A mesh topology is a 2-dimensional grid in which nodes at opposite edges are connected, so that all nodes are topologically equal. An Internet-derived topology [1] is derived from the Internet AS connectivity graph, and has long-tailed distribution of node degree.

Given a network topology, we randomly select a node to be the ispAS and attach an originAS to it (Figure 1). Before the simulation starts, every node learns a stable route to the originAS. We then repeatedly fail and recover link $[originAS, ispAS]$, causing originAS send alternate route withdrawals and announcements to ispAS. We call a pair of a withdrawal and its following announcement a *pulse*. After some number of n pulses, the link fully recovers and the originAS stops flapping. Note the final update from originAS is always a route announcement.

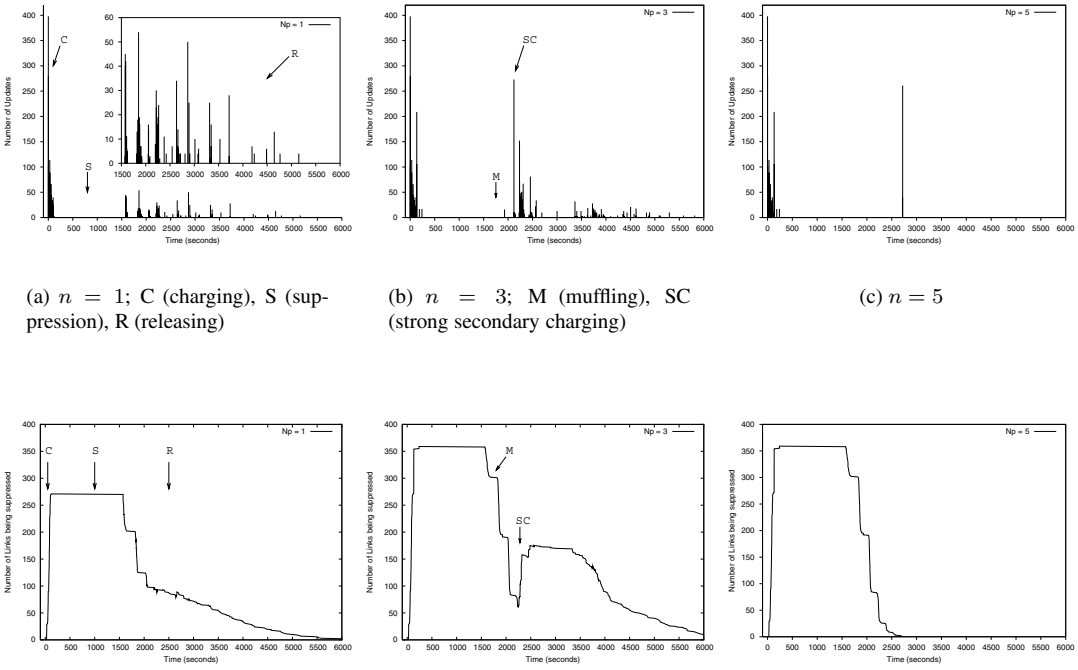
Results presented in this section are obtained from simulations with Cisco default parameters, flapping interval 60 seconds, topology size of 100 nodes, and damping enabled at all nodes. In [15], we report more simulation results from using different damping parameters, flapping intervals, topology sizes, and partial deployment of damping. Though varying different factors results in different values of convergence time and message count, the overall trend is the same as the results presented here and can be explained by our analysis in Section 4.

5.2 Simulation Results

Figures 8 and 9 show the convergence time and message count versus the number of pulses. When there is no damping, convergence time is short and the message count increases linearly with the number of pulses. For comparison purpose, we also plotted the intended behavior of convergence time based on the equations in Section 3. For intended behavior, when the number of pulses $n = 1$ or 2, route suppression is not triggered and the convergence time is the same as that of no damping; when $n \geq 3$, route suppression is triggered and the convergence time goes up. This added convergence delay is the price that damping is willing to pay for reducing message count in the network. It is determined by the originAS' flapping pattern and the ispAS' damping configuration, regardless of any other conditions in the network.

For the actual behavior, the results exhibit the same trend in both mesh topology and Internet-derived topology. For a small number of pulses, the damping dynamics deviates from the intended behavior significantly with longer convergence time. But after the critical point ($N_h = 5$), the convergence time matches the calculated values very well, which verifies our analysis in Section 4.

When $n < 5$, damping causes a long convergence delay which can be close to, or even more than one hour. Such a long delay cannot be explained by path exploration alone. Based on the damping parameter settings in Table 1, a suppression time of one hour corresponds to a penalty value of 12000. Since the penalty decays exponentially, the higher the value is, the faster it decreases. A penalty value of 12000 requires a large number of updates with very short inter-arrival time. However, path exploration cannot generate updates in such rate, because it triggers route suppression, which will block the propagation of future updates! In simulations we never observed any penalty value close to 12000. However, the long convergence time can be easily explained by secondary charging. As shown in Figure 7, path exploration charges the initial penalty value to about 3000, but since secondary charging increases the reuse timer *multiple* times later, the total convergence time can be prolonged to more than one hour.



(a) $n = 1$; C (charging), S (suppression), R (releasing)

(b) $n = 3$; M (muffling), SC (strong secondary charging)

(c) $n = 5$

(d) $n = 1$; C (charging), S (suppression), R (releasing)

(e) $n = 3$; M (muffling), SC (strong secondary charging)

(f) $n = 5$

Figure 10. Update Series and Damped Link Count

5.3 Charging, Suppression, and Releasing

We now examine the simulation results with 100-node mesh topology (Figure 10) in detail to illustrate how reuse timer interactions influence damping dynamics and cause distinct charging, suppression, and releasing periods.

The Effect of Single Pulse ($n = 1$) Figures 10(a) and 10(d) plot update series and damped link count triggered by a single pulse, respectively. The update series shows the number of update messages observed in the network in 5-second bins; the damped link count shows the total number of links being suppressed at the moment.[2]

The originAS flaps by sending an initial withdrawal and then re-announces the route 60 seconds later. The first withdrawal starts a *charging period* that lasts through the first 120 seconds. Note that even though originAS sends only one withdrawal and one announcement, Figure 10(a) shows that this single pulse is amplified to several hundred updates in the network. This one pulse is not enough to trigger route suppression on the $(originAS, ispAS)$ link, but Figure 10(d) shows that it does trigger route suppression at roughly 275 other links in the network.

After the 120^{th} second, the update messages cease and the network enters a *suppression period* that lasts from the 120^{th} second to the 1574^{th} second. During this time period, there is no outstanding routing messages. However, many preferred routes are marked as unavailable due to suppression. Finally at the 1574^{th} second, these reuse timers begin to expire and the network enters the *releasing period*. As the previously suppressed routes become available, new updates are triggered and the damped link count decreases. The releasing period lasts until the 5147^{th} second when the last update is observed.[3]

Note that although route suppressions happen in a relatively short charging period, the release of all reuse timers is spread over a long period of time. The releasing period accounts for about 70% of total convergence time and 30% of total message count. Path exploration is the cause of false suppression in the charging period, but it is the secondary charging that is responsible for the extended releasing pe-

[2]When a node suppresses routes from a neighbor node, we count it as one "damped link." Since there are 200 links in the topology, and each link can be suppressed by either end, the upper bound on damped link count is 400.

[3]Some reuse timers expire after the 5147^{th} second, but they are silent and do not contribute to either convergence time or message count.

riod. Our further examination of the simulation results confirms this claim. At first, a large number of reuse timers expire in roughly the same time period, as shown by the rapid drop of damped link count between the 1574^{th} and the 2000^{th} second. The expirations of these timers trigger a new wave of updates, which increases damping penalty on other links. As a result, some reuse timers that have not expired are postponed. Figure 7 is a typical example. Overall, secondary charging can occur multiple times, causing some reuse timers to be postponed again and again, which stretches the releasing period and exacerbates convergence time.

The Effect of Three Pulses ($n = 3$) Under our simulation settings, the third pulse will trigger suppression on the $[originAS, ispAS]$ link. As a result, the destination becomes unreachable. Comparing the Figures of $n = 1$ and $n = 3$, reuse timers that expire between the 1575^{th} and the 1927^{th} second are noisy timers in the case of single pulse, but are silent ones in the case of three pulses, which is the result of the muffling effect. Reuse timers that expire before RT_h are all muffled. The expiration of RT_h triggers a powerful secondary charging at the 1927^{th} second. When some other reuse timers expire shortly after the 2000^{th} second, both the message count and damped link count surge to a high level. The impact is so powerful that a new suppression period is formed in Figure 10(e).

The Effect of Five Pulses ($n = 5$) After five pulses, the reuse timer at ispAS (RT_h) has been increased to the point where it becomes the last timer to expire in the entire network. Beginning around the 1500^{th} second, reuse timers in the rest of the network begin to expire. Due to the muffling effect, all routers have declared the destination unreachable and these reuse timers expire silently. The last timer to expire is RT_h. When this timer expires, it triggers a route announcement sent to the network. As the route announcement propagates throughout the network, it creates a small surge of updates, but no secondary charging since there is no other pending reuse timers. For any number of pulses $n \geq 5$, the convergence time is solely determined by when RT_h fires, exactly the intended behavior predicted from the single router view of damping algorithm.

6 RCN-Enhanced Damping

Our work and [9] clearly show that when the number of flaps is small, damping can cause unintended long convergence delay. [9] proposes "selective route flap damping," in which a router attaches with each announcement a relative preference value compared with previous announcement. Based on this additional information, the receiving router estimates whether incoming updates are due to path exploration or not, and if yes, the damping penalty will not be increased. However, selective route flap damping does not detect all path exploration updates and does not address the problem of secondary charging.

Secondary charging occurs when routers far away from the flapping origin suppress routes longer than routers close to the flapping origin. In our simulations, this scenario is typically caused by path exploration. However, other factors, such as diverse damping parameter settings, can also lead to such scenario and cause secondary charging. For example, assume router Y in Figure 1 has set more aggressive damping parameters than router X, i.e., for the same sequence of updates, Y suppresses the route longer than X. After the originAS sends out a number of flaps, route suppression is triggered at both X and Y. Even if X and Y receive exactly the same number of updates with same intervals, due to their different damping parameters, X will reuse its route to originAS earlier than Y. When X reuses its route and sends it to Y, this announcement will re-charge Y's reuse timer on link $[X, Y]$.

The fundamental problem is that current damping increases the penalty value for all received updates, regardless of their root cause. Routing updates can be triggered by many different reasons, including route flapping, path exploration, route reuse, and so forth. When updates are produced by path exploration, false suppression can occur; when updates are produced by route reuse, secondary charging can occur. The damping penalty should apply only to updates caused by route flapping.

We propose to use *Root Cause Notification* (RCN) [11][2] to help guide damping decisions. RCN attaches the root cause information to each update and thus allows routers to associate each update with a particular route flap (or other cause). Each route flap (not each update) increases the damping penalty. We first review the RCN concept and then show RCN enables damping to behave as intended.

6.1 Root Cause Notification (RCN)

RCN attaches to each routing update the root cause that triggers the update. A root cause is defined as $RC = \{[u\ v], status, seq\_num\}$, where $[u\ v]$ is the root cause link, $status$ indicates whether the link is *down* or *up*, and seq_num is the sequence number associated with the link to denote the order in which root causes are generated. A node that detects the status change of an adjacent link sends out a routing update with RC attached as an additional attribute. When a node's best path changes due to the receipt of an update message, this node will copy the root cause from the incoming update into the outgoing update message. This ensures that any update that is triggered by the same link status change will carry the same root cause information.

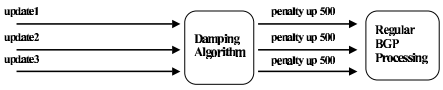


Figure 11. Damping Without RCN

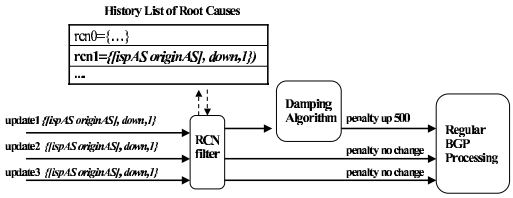


Figure 12. Damping With RCN

For example, in Figure 1, assume the current sequence number for link $[ispAS\ \ originAS]$ maintained at ispAS is 0. When ispAS first detects the failure of link $[ispAS\ \ originAS]$, it will attach a root cause of $\{[ispAS\ \ originAS], down, 1\}$ to the withdrawal triggered by this failure. All messages triggered by the same link failure will carry exactly the same root cause $\{[ispAS\ \ originAS], down, 1\}$ when they are propagated throughout the network. When link $[ispAS\ \ originAS]$ is up, ispAS will sends an announcement with root case $\{[ispAS\ originAS], up, 2\}$.

RCN was originally developed to reduce BGP slow convergence. The details of the algorithm, message overhead, and incremental deployment issues are addressed in [11]. In this paper, we make use of the RCN concept to improve damping only. More specifically, we only assume that RCN information is attached to the update messages; we do not assume that RCN is used to influence routing decisions or reduce convergence time. If RCN is used for both damping and convergence improvement, our results should remain essentially the same, except that the overall message count associated with a flap should also be reduced as a result of RCN's role in improving convergence.

6.2 Damping with RCN

With root cause information attached to each routing update, we increase damping penalty only for updates caused by route flaps. For each peer, a router maintains a recent history of root causes that have been received from that peer. When an update is received, its root cause is checked against the history list. If the root cause is already present in the history list, this update does not result in any penalty increment. If this root cause has not been reported before, a penalty increment is applied according to damping configuration, and the root cause is added into the history list. In other words, our approach acts as a filter in front of the

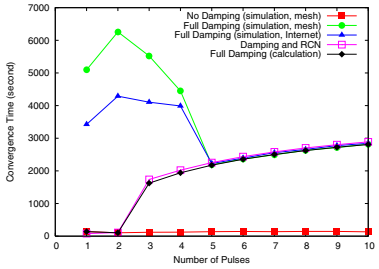


Figure 13. Convergence Time

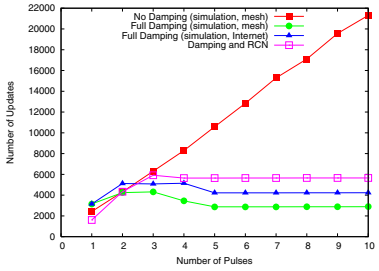


Figure 14. Message Count

damping mechanism. Even though a single route flap may lead to multiple updates, only one update is passed through the filter to the damping algorithm. Note that the filter only prevents some updates from reaching the damping algorithm; all updates are still accepted and passed to the routing decision process.

Figures 11 and 12 illustrate how the RCN helps damping. In Figure 11, a single route flap combined with path exploration results in several updates. Each of these updates adds to the damping penalty and the single flap may result in false route suppression. In Figure 12, the same sequence of updates is received but each update carries an RCN that identifies the single route flap. With this enhancement, false suppression and reuse timer interaction are prevented. Although there are many updates during path exploration, they all carry the same root cause and thus only one of these updates passes the filter and increases the damping penalty. When a suppressed route is reused, the RCN is attached to the route announcement, which will not cause penalty increase at receiving routers since the root cause have been seen before.

Figures 13 and 14 show the simulation results of RCN-enhanced damping in the 100-node mesh topology. With the help of RCN, routing convergence does not experience extra long delay when the number of pulses is small and it closely matches the calculated (intended) convergence time. At the same time, damping achieves its goal of limiting the message count when the number of pulses is large. Inter-

estingly, damping with RCN produces slightly more messages than damping without RCN. This is because when RCN is used, route suppression happens after three pluses, exactly as specified by the damping algorithm and parameters. Without RCN, false suppression happens earlier due to path exploration and reduces the number of messages. Overall, by making use of root cause information, we can prevent complex interactions in the network from having negative impact on damping, and make damping work as the design intends.

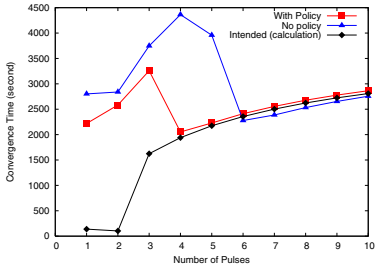


Figure 15. Impact of Policy

7 Discussion

Our analysis and simulation show that the behavior of a network is often influenced by unexpected interactions among its components. In the case of BGP route damping, path exploration results in false suppression, and unexpected timer interactions prolong the routing convergence. Our study discovered the previously unknown reuse timer interactions.

Although our analysis and simulation are based on the BGP damping mechanism, potential interactions among other components in BGP must be considered before one applies our results to infer Internet routing performance. For clarity of analysis, in this paper we only presented results using a fixed flapping interval, full damping deployment with consistent damping parameters at all the routers, and shortest path routing policy. In reality unstable destinations exhibit different flapping patterns and different routers have inconsistent damping parameter settings. Furthermore, routing policies can have a big impact on path exploration which in turn can affect the damping behavior.

For example, routing policies other than shortest path are widely used to regulate BGP route selection and propagation. BGP policies consider factors such as the commercial relationship between two networks and, as a result, some physically plausible paths may be prohibited by policy and not announced to peers. Routing policies lead to reduced number of alternate paths that can be explored during convergence period, which in turn reduces the number of routers that turn on false suppression, the main factor that sets up secondary charging. Furthermore, the shortest path policy assumes that if a route is selected after a reuse timer expires, this new route will be announced to all the neighbors. However under different routing policies, this route may not be announced to some, or even all, the neighbors. If a route is announced, the result is a noisy reuse timer; if policy forbids the announcement, an otherwise noisy reuse timer would be silenced.

To quantify the impact of routing policy on damping dynamics, we have run simulations using the no-valley routing policy, which is widely adopted in practice [4][5]. Figure 15 shows the simulation result on a 208-node Internet-derived topology, in which every pair of connected nodes is assigned a relationship as customer-provider or peer-peer. The policy regulates route announcement to ensure that a router does not transit data traffic for a third party. That is, besides traffic originated by its own, a router will forward traffic only if the traffic is coming from its customers or destined to one of its customers. The simulation results show that this policy greatly reduces the number of nodes that turn on false suppression (not shown in the figure), thus reduces secondary charging and moves the convergence time closer to the intended behavior. This unexpected reduction of slow convergence by routing policy can serve as another example of unexpected interactions among different components in a large system. However routing policies do not eliminate all path explorations or muffle all noisy timers. Our simulation results show that, although secondary charging now affects a small number of nodes, the convergence delay for these affected nodes will still be much longer than what the damping design intends.

Overall, we make two observations. First, policies that are currently in place help reduce *some* of the undesired damping behaviors. This is important to consider when translating simulation results into Internet results. Second, routing policies change over time and other unexpected factors can influence routing behavior as well. Rather than counting on routing policies to reduce slow convergence, hence reduce the negative effect of damping, we believe the RCN-enhanced damping provides the correct solution. It solves the fundamental problem by correctly identifying the route flap that triggers a particular update and applying the damping penalty to the flap itself (as opposed to the perceived result of a flap).

8 Conclusion

Route flap damping is a seemingly simple mechanism to prevent the instability of any individual route from overloading the global system. If one examines the effect of BGP's damping mechanism at a *single* router, the result is simple: all updates of a route are propagated without delay

as long as the associated penalty value is below the threshold; otherwise they are blocked. However when damping is applied to *a network of routers*, not only slow convergence can falsely trigger suppression *elsewhere* other than at the router adjacent to the flapping origin, but route reuse timer interactions can also lead to "after shock" effect, unsettling routing changes long after the flapping origin has stabilized.

Our result explains how the number of flaps affects the damping dynamics in a network of routers. For a persistently unstable route, the router closest to the flapping origin can damp the route and effectively isolate the instability from the global system, achieving the goal of the damping design. However when a route changes only a small number of times, the combined results of path exploration and secondary charging can lead to prolonged routing convergence delay. We have proposed a simple solution that can effectively eliminate false suppression due to path exploration and reuse timer interactions.

The global Internet routing infrastructure is a complex system and it is difficult or even impossible to capture all the factors that may present in the Internet routing in simulation experiments. Consequently, our simulation results may not be directly applicable to predicting BGP damping dynamics in the Internet. In particular, we believe that the commonly used no-valley routing policies can reduce the number of alternative paths, hence reducing false suppressions due to path exploration, consequently the chances for reuse timer interactions. However, due to the Internet's large scale and diversity in routing policy and damping deployment, path explorations of various degrees do exist in various parts of the Internet, providing ready conditions for reuse timer interactions, which can explain the known measurement results of prolonged routing convergence delay.

Despite its unintended behavior under certain conditions, BGP damping serves as the last fence of defense against unstable routes when other mechanisms have failed. We firmly believe that damping is a necessary mechanism to protect the global Internet routing infrastructure from melting down under high routing dynamics, and damping is equally essential to other distributed systems where resource constraints such as power, bandwidth, and router resources are limited.

The intriguing interplay of false damping and reuse timer interactions discovered in this work serves as a good example of a more general challenge: one cannot predict the resulting behavior of a protocol in a large system by examining its operation at a single component in isolation, because such studies overlook essential (and often unexpected) interactions that are inherent in a large system.

9 Acknowledgment

We would like to thank Morley Mao and BJ Premore for the help on damping simulation in SSFNet, Xiaoliang Zhao for the discussion on early version of the paper, and anonymous reviewers for their comments.

References

[1] BJ Premore. Multi-as topologies from bgp routing tables. http://www.ssfnet.org/Exchange/gallery/asgraph/index.html.

[2] J. Chandrashekar, Z. Duan, Z.-L. Zhang, and J. Krasky. Limiting path exploration in path vector protocols. In *Proc. of IEEE INFOCOM*, March 2005.

[3] G. Huston. Commentary on inter-domain routing in the Internet. RFC 3221, IETF, December 2001.

[4] Geoff Huston. Interconnection, peering and settlements, part i. *Internet Protocol Journal*, 2(1), 1999.

[5] Geoff Huston. Interconnection, peering and settlements, part ii. *Internet Protocol Journal*, 2(2), 1999.

[6] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet Routing Convergence. In *Proc. of ACM SIGCOMM*, August 2000.

[7] C. Labovitz, G. Malan, and F. Jahanian. Origins of Internet Routing Instability. In *Proc. of IEEE INFOCOM*, march 1999.

[8] Craig Labovitz, G. Robert Malan, and Farnam Jahanian. Internet routing instability. *ACM/IEEE Transactions on Networking*, 6(5):515–528, October 1998.

[9] Z. M. Mao, R. Govindan, G. Varghese, and R. Katz. Route flap damping exacerbates Internet routing convergence. In *Proc. of ACM SIGCOMM*, August 2002.

[10] Christian Panigl, Joachim Schmitz, Philip Smith, and Cristina Vistoli. Ripe routing-wg recommendations for coordinated route-flap damping parameters. RIPE 229, RIPE, October 2001.

[11] D. Pei, M. Azuma, D. Massey, and L. Zhang. BGP-RCN: Improving BGP Convergence Through Root Cause Notification. *Computer Networks*, 2005. To appear.

[12] Y. Rekhter and T. Li. Border Gateway Protocol 4. RFC 1771, Internet Engineering Task Force, July 1995.

[13] SSF Research Network. Ssfnet. http://www.ssfnet.org.

[14] C. Villamizar, R. Chandra, and R. Govindan. Bgp route flap dampening. RFC 2439, IETF, November 1998.

[15] B. Zhang, D. Massey, and L. Zhang. Bgp dynamics during route flap damping. Technical Report 03-805, USC-CSD, November 2003.