BGP Dynamics during Route Flap Damping

Beichuan Zhang
bzhang@isi.edu

Daniel Massey
masseyd@isi.edu

Lixia Zhang
lixia@cs.ucla.edu

***Abstract*—**

The BGP routing protocol uses a mechanism called *Route Flap Damping* [13] to limit the impact of connectivity instability to any individual sites. Although it is believed that this damping mechanism contributes to the stability of global Internet routing, the exact effects of damping has not been thoroughly examined in a large scale network setting. Previous work [8] has shown that damping can be falsely triggered by BGP's path exploration and significantly extends the routing convergence time after even a single route flap. In this paper we examine the impact of damping under a range of connectivity flapping patterns and different damping parameters. Our results show that damping can confine global routing dynamics to follow a predictable analytical model when connectivity to a destination flaps persistently. However when the number of flaps is small, the global routing behavior deviates from the intended analytical model and damping leads to higher dynamics as measured by both message overhead and network convergence. Such dynamics are largely shaped by the interaction between route reuse timers at different routers; route suppression and reuse at one router can affect the number of routing updates received by other routers, and in turn, others' damping behavior. We show how this reuse timer interaction, when combined with BGP path exploration, can lead to a staged behavior of routing updates consisting of charging, suppression, releasing, and possible additional rounds of secondary charging phases. We also examine the effects of flapping interval, damping parameters, and network topology on both message overhead and network convergence.

I. Introduction

As the de-facto global routing protocol for the Internet, BGP is used by thousands of Autonomous Systems (AS) to exchange routing information for hundreds of thousands of destinations (represented by IP address prefixes). In such a large-scale distributed system, faults are bound to occur and BGP routes dynamically adapt to changes in network topology and routing policy. However, a single unstable BGP route can result in thousands of update messages being propagated throughout the global Internet, increasing both router CPU load and link bandwidth consumption [6]. To help limit the impact of unstable routes, BGP employs two mechanisms to constrain the behavior of route changes. On the time scale of seconds, an MRAI timer at each sender adds a minimum time interval between two successive update messages. On a longer time scale, *Route Flap Damping* [13] uses an adaptive timer at the receiving end to constrain routing updates of unstable routes. It is believed that route flap damping has played an essential role in stabilizing the Internet routing infrastructure [3]. However a systematic in-depth study of its effectiveness on large scale networks is largely missing and the actual routing behavior under damping is not well understood.

In route flap damping, a BGP router maintains a penalty value for each prefix advertised by each peer and suppresses unstable routes based on this penalty value. Whenever the peer announces a route change, (e.g., replaces the existing route by a new one, withdraws a route, or re-announces a route after a withdrawal), the penalty is increased according to the type of the change. When the penalty value for a prefix exceeds a preset threshold, the associated peer can no longer be selected as the best path to the prefix. Although further updates may be received from the suppressed peer, these updates are not propagated further. The penalty value decays exponentially over time. When it reaches a predefined reuse threshold, the route will be considered for best path selection again. Throughout this paper, we use *damping* as an abbreviation for "route flap damping" to refer the whole mechanism, and route *suppression* to refer the specific action of stopping using a route. A more detailed description of damping and its background is given in Section II.

This paper presents a systematic study of BGP convergence delay and message overhead under various damping settings including the number of route flaps, the flapping interval, and different sets of damping parameters. Although BGP's path exploration process is the driving force behind the routing dynamics, the specific pattern of dynamics is largely shaped by a previously unknown interaction among reuse timers at different routers. Our results show that the entire damping process comprises of three distinct periods, and each period has its own characteristics. We provide a mathematical prediction of damping behavior which matches well to the simulation results when the number of flaps is high. However, when the number of flaps is low, damping behaves counter to intended design. The route suppression and reuse at one router may affect the number of updates received by other routers, and in turn, others' damping behavior. As a result, damping causes both higher convergence time and higher message overhead when the number of flaps is low. We analyze this interaction in detail, and examine the impact of flapping interval, damping parameters and network topology. Our results also show that, contrary to intuition, certain damping parameter tunings recommended by RIPE-229 [10] have *negative* effects on both convergence time and message overhead. These findings provide new insights into the damping mechanism.

The remainder of the paper is organized as follows. Section II describes damping in more detail and Section III describes the simulation methodology and setup. Section IV presents the simulation results on damping behaviors and the impact of various factors. Section V reviews related work. We conclude in Section VI with discussion and future work.

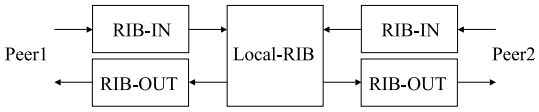


Fig. 1. A BGP router's RIBs

II. Background on Damping

BGP route flapping can be caused by various faults in hardware, software and operations. For example, route flapping may be triggered by a faulty link that intermittently fails and then recovers. In another example, the TCP connections between BGP peers may have been disrupted by the high-volume data traffic during the recent Internet worm attacks. If the BGP route to a prefix relies on the faulty link or the unstable peering session, the route will flap as the connectivity changes, and the goal of damping is to limit the global impact of such unstable routes. The damping algorithm (described below) was first implemented in 1995 and then documented in RFC 2439 in 1998 [13]. It is supported in commercial products from all major vendors. Damping is thought to be widely deployed and helps stabilize the Internet routing infrastructure [3].

Conceptually, a BGP router stores routes received from its peers in RIB-IN, picks the best route from among all the RIB-INs, stores this best route in Local-RIB, and puts updates to be sent in RIB-OUT (Fig. 1). Damping associates a penalty value with each entry in a RIB-IN. In other words, there is a penalty value associated with each peer and prefix pair. Whenever a peer sends an update for the prefix, the RIB-IN entry is updated and its corresponding penalty is also increased. Different types of updates (route change, route withdrawal, re-announcement, etc.) cause different penalty increments. If the penalty exceeds a *cut-off threshold*, the RIB-IN entry is suppressed and can no longer be used when selecting the best route. The penalty value also decays exponentially and a suppressed route will be reused when the penalty drops below a *reuse threshold*. More formally, if the penalty is $p(t_0)$ at time t_0 and $p(t)$ at time t, then

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

where λ is often configured by *half-life* $H = \ln 2/\lambda$.

A router usually sets a *reuse timer* based on current penalty value, and reuses the route when the reuse timer expires. During route suppression, if more route changes are received, RIB-IN and the penalty value will be updated accordingly, and the reuse timer will be reset based on the new penalty value. However, changes to the suppressed route do not enter Local-RIB or any RIB-OUT and thus changes to the suppressed route will not propagate further. There is also a *maximum hold-down time* as the upper limit on how long a route can be suppressed after it becomes stable. It is often implemented as an equivalent maximum penalty value. Table I lists the default damping parameters from two major vendors, and Fig. 2 illustrates the penalty value changing over time.

The operator community has long recognized that inconsistent damping settings in the network may lead to connectivity problems that are difficult to diagnose. RIPE-229 [10] recommends a set of damping parameters for router configuration. As early as in 1998, Panigl [10] observed from his operation experience that one route withdrawal and one route re-announcement in Europe triggered route suppression in North America. The exact cause of this behavior was not fully explained until 2002, when Mao et. al. [8] discovered that, after a single route change, BGP path exploration can falsely trigger route suppression at remote places. Such unexpected damping phenomena can happen in a topology as small as a 5-node clique. However, our study shows, although BGP's path exploration process is the driving force behind the routing dynamics, the specific pattern of dynamics is largely shaped by a previously unknown interaction among reuse timers at different routers.

Damping Parameters	Cisco	Juniper
Withdrawal Penalty (P_W)	1000	1000
Re-announcement Penalty (P_{RA})	0	1000
Attributes Change Penalty (P_A)	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 I
Default Damping Parameters

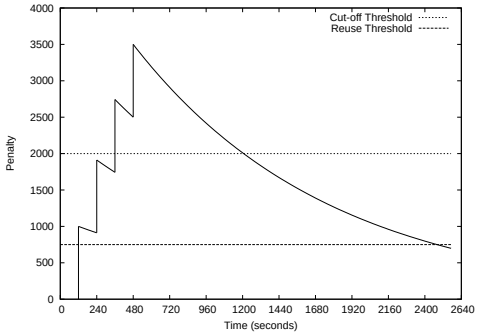


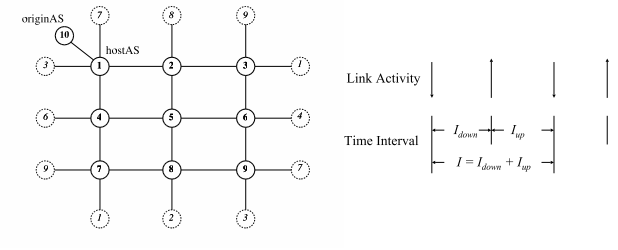
Fig. 2. Damping Penalty ($I_{down} = I_{up} = 60s$, Cisco)

One of our observations that plays an important role in this work is that there are two types of route reuse timer expiration events:

- *Noisy expiration*, which triggers some routing updates. This is because the route being reused becomes the best route, and changes the Local-RIB and RIB-OUT.
- *Silent expiration*, which does not trigger any routing update. This is because the route being reused is not the best route and makes no change to Local-RIB or RIB-OUT.

III. Simulation Methodology

We use two types of network topologies in the simulation: mesh and Internet-like. A mesh topology is a 2-dimensional grid in which nodes at opposite edges are connected (Fig. 3(a)). All nodes in a mesh are topologically equal. An Internet-like topology [1] is derived from the Internet inter-AS connectivity graph, and has long-tailed distribution for node degree. Using



(a) 3 x 3, degree = 4 (b) flapping interval

Fig. 3. Sample topology and flapping interval

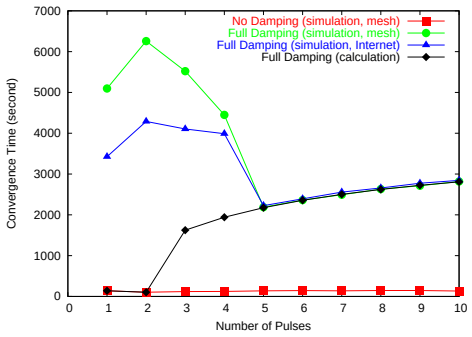


Fig. 4. Convergence Time

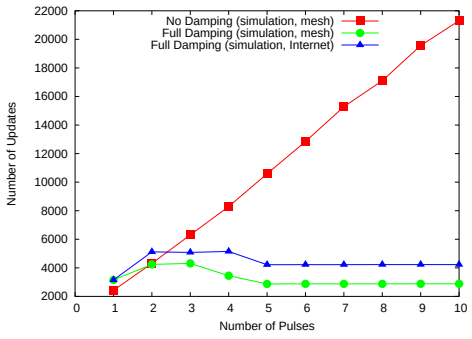


Fig. 5. Message Overhead

the mesh topology enables us to vary network size and node degree independently, while using the Internet-like topology gives more confidence on the result's applicability to the real Internet. We have tested mesh topologies ranging from 36 nodes to 900 nodes, degree 4 to 12, and Internet-like topologies ranging from 29 nodes to 830 nodes. For the ease of presentation, in this paper we show the results from a 100-node mesh topology with node degree of 4 unless otherwise stated. The impact of network size and node degree is discussed in Section IV-F.

Given a network topology, an additional node called *originAS* is attached to an existing node called *hostAS* in the network (Fig. 3(a)). Before the simulation starts, every node has a stable route to the originAS. During the simulation, the originAS sends route announcements and withdrawals alternately to the hostAS to simulate route flapping. A pair of a withdrawal and its following announcement is called a *pulse*. After a certain number of pulses, the originAS stops flapping, and its final update is a route announcement. The choice of hostAS makes no difference in mesh topologies. In Internet-like topologies, it affects the results quantitatively, but exhibits similar dynamics patterns.

We use I_{down} to denote the time interval between a withdrawal and its following announcement, I_{up} to denote the time interval between an announcement and its following withdrawal, and I ($= I_{down} + I_{up}$) to denote the time interval between two nearest withdrawals or announcements (Fig. 3(b)). In this paper, only fixed-rate flapping patterns, i.e., I_{down} and I_{up} are constants during each simulation run, are used. The default values are $I_{down} = I_{up}$ = 60 seconds. The impact of flapping interval is discussed in Section IV-D.

Two BGP performance metrics, convergence time and message overhead, are used to quantify the dynamics. The convergence time is defined as the time from when the originAS stops flapping (i.e., sending its final route announcement) to when the last BGP update message is observed in the network. The message overhead is the total number of update messages observed in the network starting from the first flap.

We model BGP as a Simple Path Vector Protocol (SPVP) [11], and conduct simulations in SSFNet [12] with our improved damping implementation. We use the default 30 seconds for MRAI timer, 2ms for link delay, 100ms for processing delay, and Cisco default damping parameters. Larger link delay (e.g., 20ms) and smaller processing delay (e.g., 10ms) give similar results. The impact of damping parameters is discussed in Section IV-E. We assume damping is enabled by all nodes and the same damping parameters are used throughout the network. The impact of partial deployment and inconsistent parameters is among our future work.

IV. BGP Dynamics during Damping

Fig. 4 and Fig. 5 show the BGP convergence time and message overhead as the number of pulses, n, increases. The shapes of these curves are typical in most simulations, including both mesh and Internet-like topologies. In this section, we explain these curves in detail, reveal the underlying causes, and study the impact of various factors.

A. Route Flapping without Damping

We first study the case that damping is turned off at all nodes to understand the effect of route flapping only.

Labovitz et. al. [5] categorize BGP convergence events into four types: T_{down} (a previously available route is withdrawn), T_{up} (a previously unavailable route is announced available), T_{long} (a route is replaced by another with longer AS path), and T_{short} (a route is replaced by another with shorter AS path). Our flapping pattern is a series of alternate T_{down} and T_{up}. Table II compares T_{down} and T_{up} against a single pulse. Confirming results from previous work, BGP's path exploration causes long convergence time and lots of update messages after T_{down}. For T_{up} and flapping, the convergence time is the time for the final route announcement to propagate through the entire network. In the case of pure T_{up}, since the network has been quiet

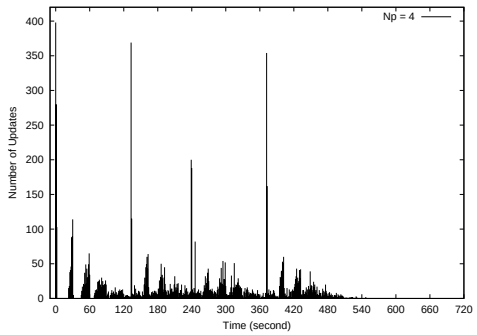


Fig. 6. Update Series (no damping, $n = 4$)

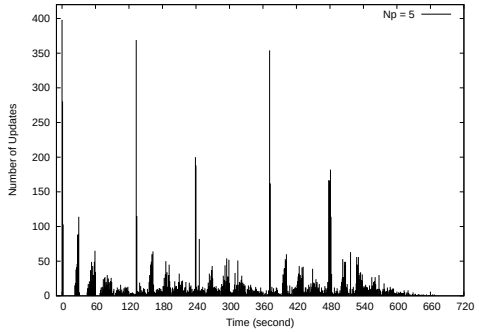


Fig. 7. Update Series (no damping, $n = 5$)

before the event, MRAI timer does not apply, and the only limiting factors are link delay and processing delay. During route flapping, however, T_{up} happens before routing updates caused by the previous T_{down} have finished. Therefore, MRAI timer takes effect to limit the propagation rate, resulting in longer convergence time. This convergence time is mainly determined by the network topology and MRAI timer, regardless of the number of flaps. This is why we see an almost flat line in Fig. 4.

To see why flapping's message overhead increases linearly with the number of pulses n, we plot the update series for $n = 4$ (Fig. 6) and $n = 5$ (Fig. 7), showing the number of update messages observed in the network within each second. The graph is composed of withdrawal-induced spikes at every $I = 120$ seconds (i.e., the interval between two withdrawals), and small fluctuations clustered around every MRAI interval (i.e., 30 seconds). Comparing the two graphs, we can see each 120-second period exhibits similar pattern, implying that each pulse contributes similar amount of updates to the total message count independently.

	T_{down}	T_{up}	Flapping ($n = 1$)
Convergence Time	735 s	1 s	139 s
Message Overhead	12469	302	2424

TABLE II
FLAPPING V.S. T_{down} AND T_{up}

B. The Intended Behavior of Damping

According to the damping algorithm, the routing dynamics can be described by a simple analytical model. When damping is enabled, persistent flaps from the originAS will trigger the hostAS to suppress its route to the originAS. After the flapping stops, it will take time R for the penalty value p to drop below the reuse threshold P_{reuse} before the hostAS can send out the route announcement. The announcement will trigger a T_{up} event which takes time CT_{up} for the network to converge. Since usually $R \gg CT_{up}$, the total convergence time is

$$CT = R + CT_{up} \simeq R$$

$$R = \frac{1}{\lambda} \ln \frac{p}{P_{reuse}}$$

Let $w(i)$ be the time between the ith flap and the $(i-1)$th flap, and $f(i)$ be the penalty increment caused by the ith flap, $i = 1, 2, \ldots, k-1, k$, and $w(1) = 0$. Right after the kth flap, the penalty value $p(k)$ will be

$$p(k) = p(k-1) * e^{-\lambda w(k)} + f(k)$$

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

In our simulation, one pulse comprises of two flaps, a withdrawal and the following re-announcement. Assuming the penalty increases by P_W for each withdrawal and P_{RA} for each re-announcement, our fixed-rate flapping pattern can be defined as

$$w(i) = \begin{matrix} I_{down} & i = 2m \\ I_{up} & i = 2m-1 \end{matrix}$$

$$f(i) = \begin{matrix} P_W & i = 2m-1 \\ P_{RA} & i = 2m \end{matrix}$$

where $m = (1, 2, \ldots)$ and $w(1) = 0$. Given these functions, we can derive the penalty value right after the nth pulse, and the result is

$$p = z * \sum_{i=1}^{n} e^{-\lambda I(i-1)} = \frac{1 - e^{-\lambda I n}}{1 - e^{-\lambda I}} * z$$

$$z = P_W e^{-\lambda I_{down}} + P_{RA}$$

Using Cisco default parameters (Table I) in above equations, we calculate the convergence time and plot it in Fig. 4. When number of pulses $n = 1$ and 2, route suppression is not triggered (Fig. 2) and the convergence time should be the same as that of no damping. When $n \geq 3$, route suppression is triggered and the convergence time should go up. This is the intended convergence time by the damping algorithm, and is the price that damping is willing to pay for routing stability. It is determined only by the flapping pattern and damping parameters, $w(i)$ and $f(i)$, at hostAS, regardless of anything else in the network.

The intended message overhead by the damping algorithm generally cannot be obtained analytically, since it depends on

the network topology and timing of updates. Nevertheless, it is expected to be almost constant when $n \geq 3$, comprising of messages before hostAS suppresses the route and messages after hostAS reuses the route, because pulses after the first two will be suppressed and not be able to cause any update in the network.

C. Route Flap Damping

1) The Basic Dynamics Pattern: In Fig. 4 and Fig. 5, simulations on both mesh topology (100 nodes) and Internet-like topology (110 nodes) exhibit the same curve shape: the simulation results match the calculated values very well when the number of pulses is large, but are substantially higher when the number of pulses is small. We call the simulation result *conformal* when it matches the intended value, and *non-conformal* when it does not. The turning point, N_h, of the curve is defined as a certain number of pulses, that when $n \geq N_h$ the curve is conformal, and when $n < N_h$ the curve is non-conformal. The existence of N_h is the basic dynamics pattern we have observed in simulations with a wide range of flapping intervals, topologies and damping parameters. In this subsection, we examine the underlying reason for this pattern.

2) Reuse Timer Interaction: Mao et. al. [8] studied the case of $n = 1$ in full-mesh topologies. They showed convincingly that even there is only one pulse, some routers in the network may receive multiple update messages due to BGP path exploration caused by the route withdrawal. Therefore route suppression may happen when $n = 1$, not at hostAS, but somewhere remotely in the network. The network will not converge until the route is reused, and this exacerbates the convergence time.

However, even after taking into account path exploration, the convergence time at $n = 1$, more than 3000 seconds in Fig. 4, is still too high. Since the penalty decays exponentially, suppressing a route for longer than 3000 seconds requires a very high penalty value, namely a large number of updates between two peers, which is not likely to be done by path exploration within $I_{down} = 60s$. Besides, path exploration alone cannot explain the curve for $n \geq 2$ either. There must be another force affecting the dynamics. We discovered that the missing factor is reuse timer interaction, a previously unknown interaction during damping.

To explain the reuse timer interaction, we plot update series and damped link count for $n = 1, 3, 5$ (Fig. 8). The update series shows the number of update messages observed in the network within every 5 seconds; the damped link count shows the total number of links being suppressed in the network. Since there are 200 links in our 100-node mesh topology, and each link can be suppressed by either end, the upper bound on damped link count is 400. We now discuss the cases of $n = 1, 2, 3, 4, 5$ one by one.

$n = 1$*:* Fig. 8(a)(d) clearly show that there are three distinct time periods during the convergence process.

- *Charging period*, from the beginning of the simulation to the 120th second, when a large number of messages are being sent and the damped link count increases rapidly, indicating that path exploration is happening. The result is that many links are suppressed and more than half of the nodes lost their routes to the originAS.

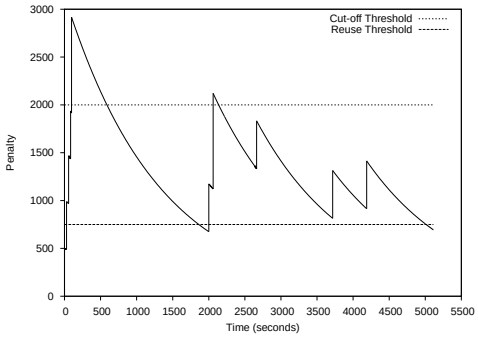


Fig. 9. Damping Penalty on Link 46-47 ($n = 1$)

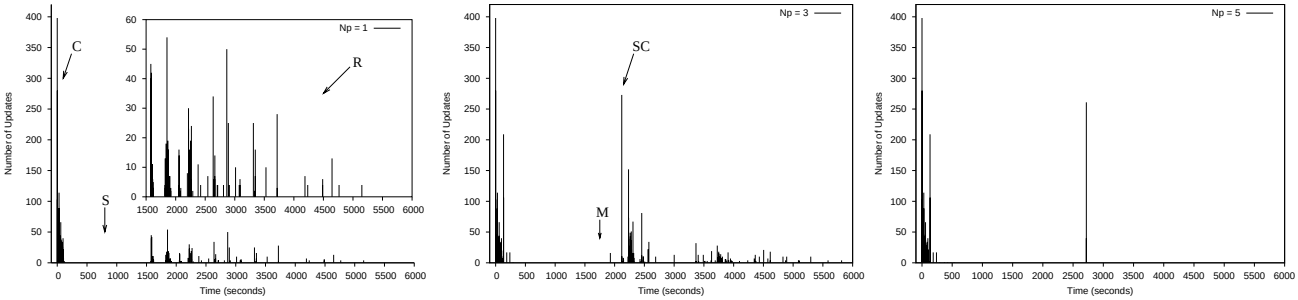
- *Suppression period*, from the 120th second to the 1574th second, when the network is quiet with no update message nor change in damped link count.
- *Releasing period*, from the 1574th second to the 5147th second [1], when previously suppressed routes are being reused as their reuse timers expire.

Unlike that all route suppressions happen in a relatively short charging period, the release of all reuse timers takes long time. The releasing period accounts for about 70% of total convergence time and 30% of total message count. This is the part that path exploration cannot explain. Further examination of simulation data shows that it is the interaction among reuse timers that stretches the releasing period. At first, reuse timers expire pretty fast as shown by the rapid drop of damped link count between the 1574th and the 2000th second. However, updates generated by noisy reuse timer expiration will arrive at some other nodes and increase their damping penalty. As a result, some reuse timers that have not expired will be postponed by these updates. We call this kind of interaction the *secondary charging effect*. Sometimes this effect can not only postpone existing reuse timers, but also cause new route suppressions. We pick a long-lasting reuse timer and plot its penalty over time in Fig. 9. After the charging period, the penalty decays smoothly to below the reuse threshold. But soon it is pushed back to above cut-off threshold again by the secondary charging effect. Before the route is eventually reused, the penalty is pushed back by another three times. Secondary charging may happen more than once, causing some reuse timers being postponed again and again. This effect stretches the releasing period and exacerbates convergence time.

$n = 2$*:* When the second pulse comes, more reuse timers are set in the charging period due to extra routing updates. During the releasing period, the secondary charging effect will be able to affect more nodes and makes the convergence time longer.

$n = 3$*:* Based on our simulation setting, the third pulse will trigger hostAS to suppress its route to the originAS. We use RT_h to denote this special reuse timer. There are two interesting observations in Fig. 8(b) and (e) about RT_h. One is that during the first part of releasing period, i.e., between the 1575th and the 1927th second, a lot of reuse timers that had noisy expi-

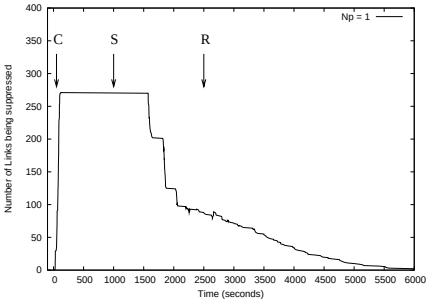
[1]Timers expired after the 5147th second are silent and do not contribute to either convergence time or message overhead.



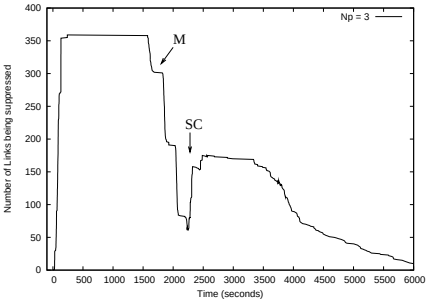
(a) $n = 1$; C (charging), S (suppression), R (releasing)

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

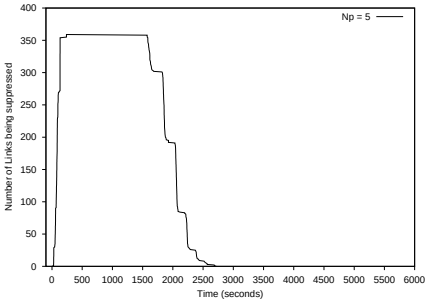
(c) $n = 5$



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



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



(f) $n = 5$

Fig. 8. Update Series and Damped Link Count (Full Damping)

ration previously now expire silently. This is due to the *muffling effect* caused by RT_h. When the hostAS suppresses its route to the originAS upon receiving the third pulse, it has no alternate route and has to send a withdrawal to its other peers. Therefore, damping does not stop withdrawals when the prefix becomes unreachable. For the same reason, this withdrawal will propagate through the entire network and eventually every node has no route to the prefix. Since all future route announcements from the originAS are suppressed by the hostAS, this situation of no route does not change until RT_h expires. Therefore any reuse timer expiration before RT_h becomes silent.

Another observation is the powerful secondary charging caused by RT_h. The hostAS reuses its route at the 1927th second, which restores the route to some nodes and removes the muffling effect. When some other reuse timers expire shortly after the 2000th second, both the message count and damped link count surge to a high level. The impact is so powerful that a new plateau is formed in Fig. 8(e), which means many new reuse timers are set. Since all other routers rely on hostAS to reach originAS, the hostAS's route reuse potentially can affect all routers, and cause more powerful secondary charging.

These two types of reuse timer interaction compete against each other: the secondary charging effect stretches the convergence time, while the muffling effect reduces the convergence time by having reuse timers expire silently. The net result in this simulation run is a somewhat shorter convergence time at $n = 3$ than $n = 2$.

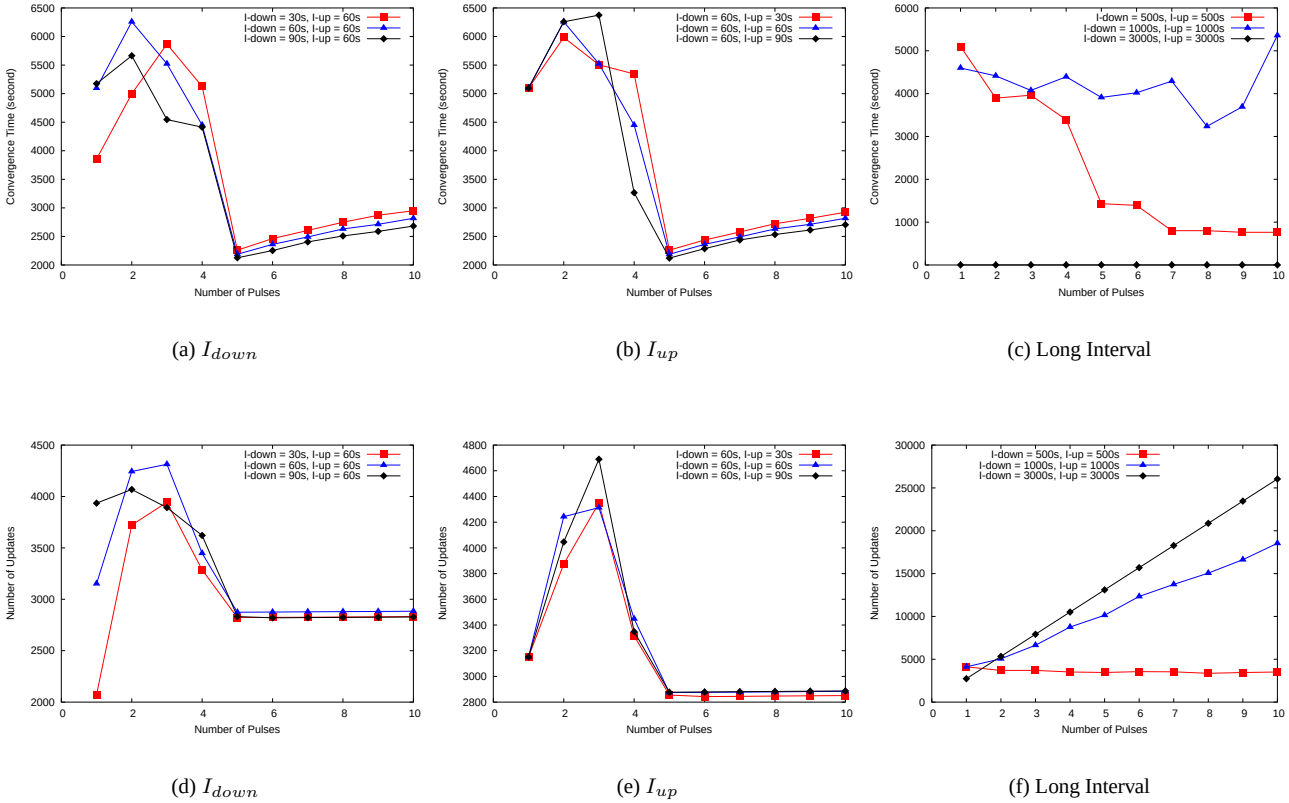
$n = 4$*:* When the originAS flaps more, additional flaps are only experienced by the hostAS. Therefore, the only effect of flaps $n > 3$ is to postpone RT_h only, while reuse timers in the rest of the network keep the same. Larger RT_h is able to muffle more reuse timer expiration and shorten convergence time further.

$n = 5$*:* In Fig. 8(c)(f), RT_h has been postponed for so long that it becomes the last one to expire. At the time of its expiration, all other reuse timers have expired silently due to the muffling effect, leaving no secondary charging effect but a single T_{up} event. From this point on ($n \geq 5$), the convergence time is solely determined by when RT_h fires, exactly the intended behavior of the damping algorithm, and the curve becomes conformal. Comparing graphs of damped link count in Fig. 8, (f) shows when all the reuse timers are scheduled to expire after path exploration, (d) shows the secondary charging effect stretches the expiration time, and (e) shows the expiration time is restored to the originally scheduled time due to the muffling effect.

Denoting the last reuse timer to fire in the network by RT_{net}, the curves turning point, N_h, is when

$$RT_h > RT_{net}$$

The same reasoning can be applied to explain the results on message overhead (Fig. 5). When $n < 3$, the message overhead



(a) I_{down} (b) I_{up} (c) Long Interval

(d) I_{down} (e) I_{up} (f) Long Interval

Fig. 10. Impact of Flapping Intervals

increases due to path exploration and secondary charging. After $n = 3$, the muffling effect reduces the message overhead, until $n = 5$, when $RT_h > RT_{net}$. From that point on, the message overhead becomes almost constant because it is the sum of two relatively constant parts: the message count during the charging period, and the message count caused by the last T_{up} event.

3) Summary: The convergence process during flapping and damping comprises of three periods, charging, suppression and releasing. The charging period contributes major portion of total message overhead, while the releasing period contributes major portion of convergence time. There are two types of reuse timer interaction competing against each other: the secondary charging effect stretches convergence time, but the muffling effect reduces the number of noisy timer expiration. When the number of pulses is greater than the turning point (N_h), the reuse timer at hostAS (RT_h) will outlast the last reuse timer in the network (RT_{net}), making the muffling effect dominant and bringing the convergence time and message overhead conformal with intended values.

D. Flapping Intervals

Though the flapping interval $I_{down} = I_{up} = 60s$ is chosen rather arbitrarily, the insight obtained from the simulation helps understand the dynamics caused by other interval values.

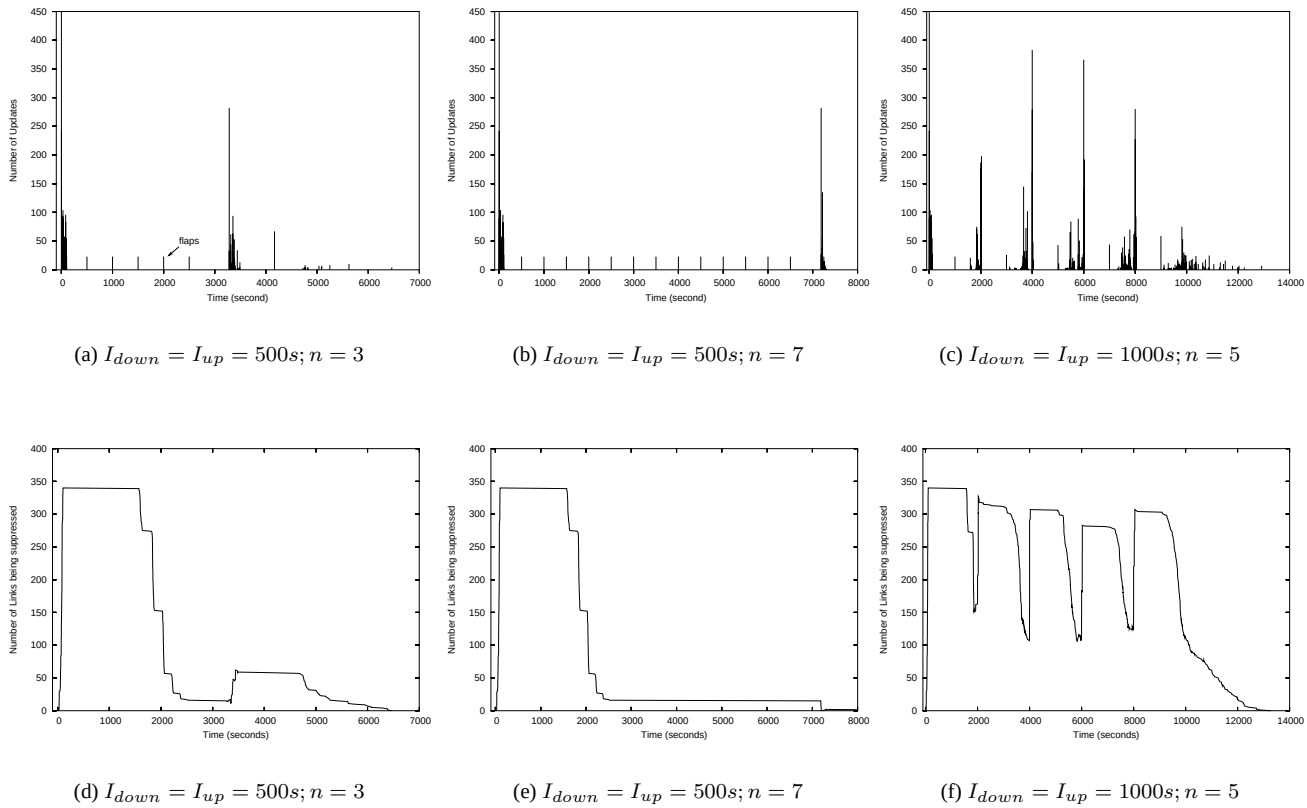
1) $I < I_s$: One important condition of the basic dynamics pattern is the existence of RT_h. According to the damping algorithm, when flapping interval is fixed, the penalty value will eventually reach an upper limit, when the penalty increment caused by one flap equals to the decay since last flap. This penalty limit decreases as the flapping interval increases, and when it is less than the cut-off threshold P_{cut}, route suppression will never be triggered at hostAS and RT_h will never be set. By solving

$$P_{cut} = p(k) = p(k-1) * e^{-\lambda w(k)} + f(k) = p(k-1)$$

we can obtain the maximum interval I_s for RT_h to be set, and with our parameters, $I_s = 900s$.

Fig. 10 (a)(b)(d)(e) show the impact of flapping intervals from 30 seconds to 90 seconds. Results on 10 seconds and 180 seconds are not shown, but similar to Fig. 10. There are two observations. First, the convergence time and message overhead at $n = 1$ are determined by I_{down} only. Longer I_{down} allows path exploration to complete more, resulting in longer convergence time and more message overhead. Second, when $n \geq 5$, the convergence time is mainly determined by $I = I_{down} + I_{up}$, as more frequent flaps cause longer convergence time. For data points in between, the exact values are up to the subtle timing of message transmission and various BGP timers, but all curves follow the same basic shape. Overall, when $I \leq I_s$, different flapping intervals change data points quantitatively, but the basic dynamics pattern holds.

2) $I_s < I < RT_h^{\varnothing}$: Fig. 10 (c)(f) show the results for $I_{down} = I_{up} = 500s$, so $I = 1000s > I_s = 900s$. Even though it is impossible for RT_h to be set, the basic shape



(a) $I_{down} = I_{up} = 500s; n = 3$ (b) $I_{down} = I_{up} = 500s; n = 7$ (c) $I_{down} = I_{up} = 1000s; n = 5$

(d) $I_{down} = I_{up} = 500s; n = 3$ (e) $I_{down} = I_{up} = 500s; n = 7$ (f) $I_{down} = I_{up} = 1000s; n = 5$

Fig. 11. Update Series and Damped Link Count (Flapping Interval)

still holds, with the turning point $N_h = 7$. This seemingly contradicting result can be explained by path exploration and reuse timer interaction too. Although hostAS will not suppress its route to originAS, some other nodes will because they receive more frequent updates due to path exploration. In fact, in this particular simulation run, route suppression is triggered at nodes two or more hops away from the hostAS. Therefore, the network has two areas: nodes close to hostAS without route suppression, and nodes away from hostAS with possible route suppression. At the boundary of these two areas, there is a set of reuse timers RT_{h1}, RT_{h2} etc., which is collectively denoted by $RT_h^{\varnothing}$. After $RT_h^{\varnothing}$ is set, additional flaps will increase $RT_h^{\varnothing}$, but will not be able to affect any other reuse timer in the network. Therefore, it is this $RT_h^{\varnothing}$ that functions similarly to original RT_h (e.g., muffling, strong secondary charging) and makes convergence time and message overhead eventually conformal.

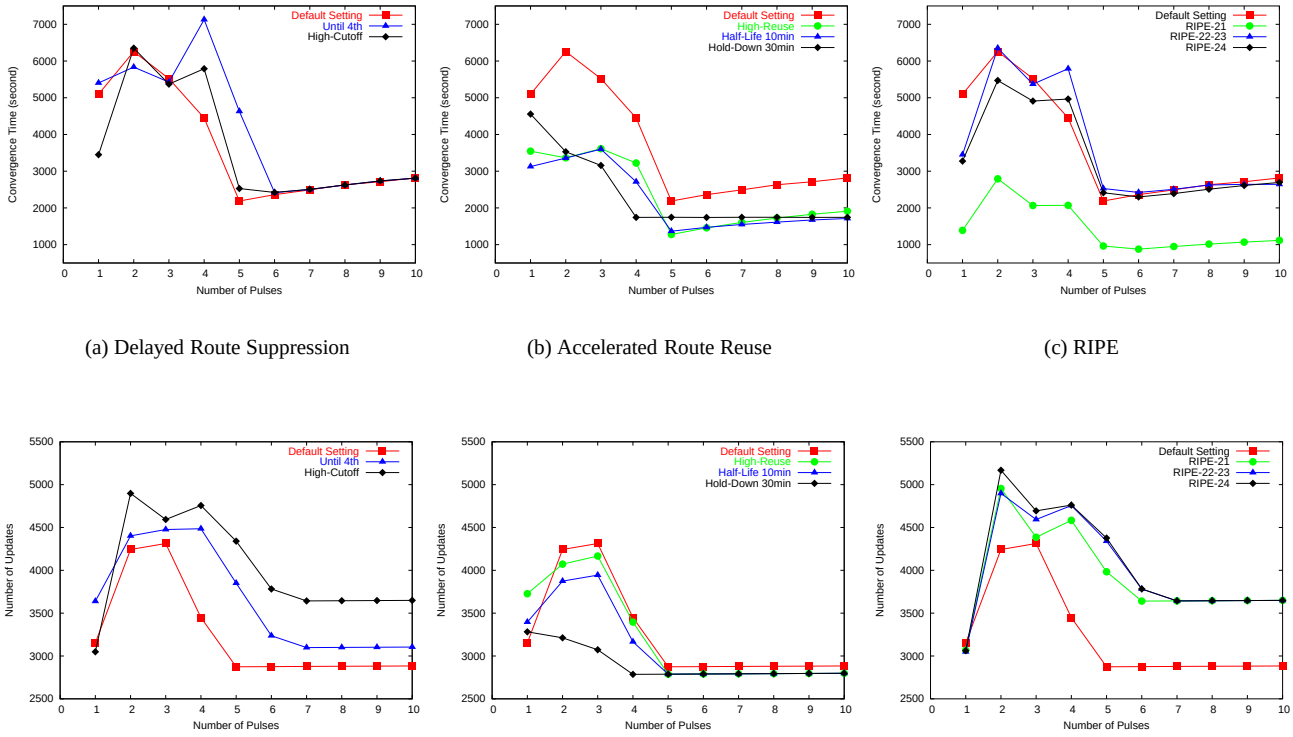
This is better explained with Fig. 11 (a)(d)(b)(e) showing the update series and damped link count. When $n = 3$, the muffling effect is apparent during the early part of the releasing period as most reuse timers expire silently. By the time of 2521st second, the remaining 16 reuse timers are all at the boundary and belong to $RT_h^{\varnothing}$. Since $RT_h^{\varnothing}$ comprises of different timers with different expiration times, its release spans a long time period. In this simulation run, it starts at the 3279th second and lasts for 204 seconds. These multiple T_{up} events issued in a relatively long time period make MRAI timer less effective in aggregating updates, causing more update messages in the network, and many reuse timers are set again as a result. Therefore, although $RT_h^{\varnothing}$ is the last to expire, it fails to make the curve conformal.

As the originAS flaps more, additional pulses help synchronize the different timers in $RT_h^{\varnothing}$. The difference between two reuse timers, RT_{h1} and RT_{h2}, is determined by their penalty values p_{h1} and p_{h2} as

$$RT_{h1} - RT_{h2} = \frac{1}{\lambda} \ln \frac{p_{h1}}{p_{h2}}$$

As shown in Fig. 6 and Fig. 7, each additional pulse causes similar number of update messages, which will add similar penalty increment to both p_{h1} and p_{h2}. Therefore, regardless of their initial values, as the originAS flaps more, $\frac{p_{h1}}{p_{h2}}$ approaches to 1 and $(RT_{h1} - RT_{h2})$ approaches to 0. At $n = 7$, the expiration time period of $RT_h^{\varnothing}$ has reduced to 15 seconds, indicated by a much more narrow peak starting at 7185th second in Fig. 11 (b), and this is good enough in our topology to have negligible secondary charging effect. From this point on ($n \geq 7$), the convergence time and message overhead become conformal.

3) $RT_h^{\varnothing} < I < T$: In this case, the $RT_h^{\varnothing}$ timer expires before the next pulse comes. Therefore, $RT_h^{\varnothing}$ will not be able to contain flaps within the boundary, and its expiration time will not be able to accumulate to a large value out-lasting other timers in the network. Since $RT_h^{\varnothing}$ is no longer effective, convergence time and message overhead will not become conformal and there is no turning point for the curve, which is clearly shown in Fig. 10 (c) (f) when $I_{down} = I_{up} = 1000s$. From its update series and damped link count (Fig. 11 (c) (f)), we can see that each pulse is relatively independent, has similar impact on



(a) Delayed Route Suppression (b) Accelerated Route Reuse (c) RIPE

(d) Delayed Route Suppression (e) Accelerated Route Reuse (f) RIPE

Fig. 12. Impact of Damping Parameters

both number of updates and damped link count, and its charging period overlaps with the releasing period of the previous pulse. After the flapping stops, there is no muffling effect to reduce reuse timer interaction during the last releasing period, and the convergence time remains at a high level.

4) $T < I$: For a single pulse, suppose the total time of its charging period, suppression period and releasing period is T. When $T < I$, the previous releasing period and the following charging period is totally separated, which means the network has converged before the next pulse comes, and each pulse acts totally independently from each other. Since we start counting the convergence time from the last flap, i.e., the final announcement, and the time between the final announcement and its preceding withdrawal is I_{down}, the convergence time becomes $(T - I_{down})$, regardless of the number of pulses. Fig. 10 (c)(f) show a special case where $I_{down} = I_{up} = 3000s$. Note T is less than 3000 seconds read from Fig. 8. Therefore the convergence time is always that of a pure T_{up}, which is 1 second from Table II.

5) Summary: In a network of only two nodes, when the flapping is frequent enough to reach the cut-off threshold, damping will take effect; otherwise it will not. However, it is far more complicated in large networks. When $I < I_s$, RT_h will be set and the basic dynamics pattern holds; when $I_s < I$, $RT_h^{\varnothing}$ functions similarly as RT_h, but its effectiveness depends on how well its individual timers are synchronized. For very large flapping intervals, even $RT_h^{\varnothing}$ can no longer make the convergence time and message overhead conformal.

E. Damping Parameters

1) Parameter Settings: The network operator community has long recognized that inconsistent damping parameters used in the network may cause connectivity problems that are difficult to diagnose. They also noticed that the vendor default parameters are sometimes so aggressive that relatively stable routes can be suppressed. RIPE has published recommendations for damping parameters [10] to encourage consistent settings and less aggressive damping. The recommendation calls for "progressive damping", which uses different parameter settings for different prefix lengths. The intuition behind this is that shorter prefixes should be suppressed less aggressively because they are likely to be more stable and represent more users. It also provides an incentive for longer prefixes to be aggregated.

There are three RIPE-recommended settings (Table III): RIPE-24 for prefix /24 and longer, RIPE-22-23 for prefix /22 and /23, and RIPE-21 for prefix /21 and shorter. Compared with the default setting, all three RIPE recommendations use less aggressive parameters, though to different degrees and with different combinations. Besides, a common change is "Until-4th", which means a route will not be suppressed until the 4th flap. The intended purpose of these parameter tunings is to avoid suppressing relatively stable routes. In this subsection, we study their actual impacts on convergence time and message overhead.

2) Individual Parameter Tuning: To see the effect of tuning each individual parameter, we change one parameter at a

Type	Half Life	Reuse	Cutoff	Max Hold Down
Default	15 min	750	2000	60 min
Until 4th *	15 min	750	2000	60 min
High Cutoff	15 min	750	**3000**	60 min
High Reuse	15min	**1500**	2000	60 min
Short Half Life	**10 min**	750	2000	60 min
Low Max Hold	15 min	750	2000	**30 min**
RIPE-24 *	15 min	**820**	**3000**	60 min
RIPE-22-23 *	15 min	750	**3000**	**45 min**
RIPE-21 *	**10 min**	**1500**	**3000**	**30 min**

TABLE III
DAMPING PARAMETER SETTINGS (* NO SUPPRESSION UNTIL THE 4TH FLAP)

Size	Degree	#Link * 2	#Msg	#Msg/(#Link*2)
100	4	400	2568	6.42
100	8	800	5456	6.82
100	12	1200	8450	7.04
400	4	1600	10002	6.25
400	8	3200	20240	6.33
400	12	4800	31274	6.52
900	4	3600	23072	6.41
110	-	572	3609	6.31
830	-	6978	48644	6.97

TABLE IV
AVERAGE UPDATE PER LINK IN DIFFERENT TOPOLOGIES DURING CHARGING PERIOD

time from the default value to the RIPE-recommended value, and end up with five settings: Until-4th, High-Cutoff, High-Reuse, Short-Half-Life, and Low-Max-Hold (Table III). These settings can be categorized into two types: Until-4th and High-Cutoff delay route suppression, while High-Reuse, Short-Half-Life and Low-Max-Hold accelerate route reuse.

Fig. 12 (a) (d) show that delaying route suppression often makes convergence time and message overhead worse than the default setting. The reason is path exploration. Even if there is only one or two pulses, path exploration has already begun. When route suppression takes effect, it actually slows down path exploration and reduces number of messages in the network significantly. If route suppression is postponed, path exploration will continue to produce more messages, which will push the damping penalty even higher and trigger route suppression soon after. Because some nodes have accumulated higher penalty value, the maximum reuse timer in the network, RT_{net}, increases too, shifting the curve's turning point higher, $N_h = 7$ for both Until-4th and High-Cutoff. Therefore, delaying route suppression by a small amount is likely to backfire instead of improving performance.

Fig. 12 (b) (e) show that accelerating route reuse often gives shorter convergence time and less message overhead. It does not affect when the route is suppressed, but makes the route available sooner. Route suppression happens as before to reduce the impact of path exploration. After the flapping stops, accelerating route reuse makes the suppression period and releasing period shorter. Therefore, all the reuse timers are scheduled to expire within a shorter time period, which makes MRAI timer more effective in aggregating update messages. As a result, there are less update messages in the network, less secondary charging effect and shorter convergence time. Take the Max-Hold-Down = 30min as an example. This maximum hold-down time translates to the maximum penalty value of 3000, a dramatic reduction from the default value 12000. When $n = 1$, its performance is not much different from the default setting because only a small number of routes have reached the maximum penalty. When $n \geq 2$, many routes have reached the maximum penalty 3000 and have very similar expiration times. Some of those expire earlier are subject to muffling effect. Therefore, all noisy timer expirations happen within a relatively short time period, and the MRAI timer is able to reduce the number of messages significantly. The convergence time is shorter too because of less secondary charging effect.

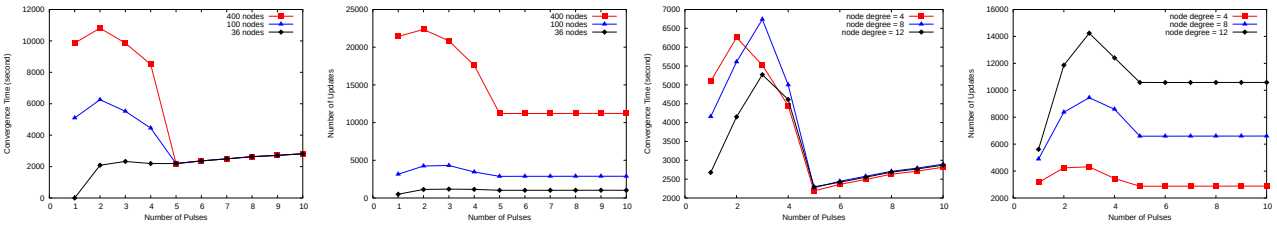
3) RIPE-recommended Settings: The RIPE recommendations adopt Until-4th in all three settings, plus a mix of other tuning. However, our study shows that delaying route suppression and accelerating route reuse have almost opposite effects. As a result, only RIPE-21 clearly shortens convergence time, while the other two have similar convergence time compared with the default setting. All three recommended settings have more message overhead. (Fig. 12 (c)(f)). Our study suggests that it would fare better if only tunings accelerating route reuse have been used.

F. Topology

Reuse timer interaction does not happen in all types of topologies. The sufficient condition of reuse timer interaction is having a noisy expiration on one's most preferred path to the flapping source, and path exploration is a very effective way to create this situation in a large network. However, in extreme topologies like clique (full-mesh) and single-line, this condition does not hold. In a clique, since every node is directly connected to the originAS, the final route announcement from the originAS will arrive at every node's RIB-IN immediately. Once this best route is reused and installed in everyone's Local-RIB, the network converges, and no other route reuse can affect it. In a single-line topology, there is no path exploration. Therefore route reuse interaction will not happen if consistent damping parameters are used in all nodes in a single-line topology.

Nevertheless, for general topologies in between, path exploration exists and route reuse timer interaction is likely to happen. This is confirmed by our simulations on both mesh and Internet-like topologies. In this subsection, we vary the network size and node degree in the mesh topology to study the impact of these topological factors. The results are shown in Fig. 13.

In the topology of 36 nodes, path exploration is negligible when $n = 1$, as both convergence time and message overhead are very low. At $n = 2, 3, 4$, path exploration and reuse timer



(a) network size (b) network size (c) node degree (d) node degree

Fig. 13. Impact of Topology

interaction take effect to increase convergence time and message overhead in all sizes. As the network size increases, the number of reuse timers being set in the network increases too, making more interactions possible. Therefore it is more likely that the secondary charging happens multiple times, and affects more nodes. This stretches the convergence time much longer However, the curve's turning point is still at $N_h = 5$ for all sizes.

As node degree increases, the convergence time decreases at $n = 1$. This is because larger node degree means smaller network diameter, so that the re-announcement can reach all nodes quickly to cancel the on-going path exploration, resulting in less reuse timers in the network. But at $n = 3$, the last re-announcement is suppressed by the hostAS since the last withdrawal has triggered route suppression at hostAS. This last withdrawal will trigger a full path exploration, which causes more reuse timers for higher node degree due to more alternate paths. Therefore the convergence time peaks at $n = 3$ instead of 2 for degree of 8 and 12. The curve's turning point is still at $N_h = 5$ for all node degrees.

As network size and node degree increase substantially, more alternate paths lead to heavier path exploration, which will cause more messages and route suppressions. It was expected that the turning point N_h will increase since RT_h would need more flaps to outlast all reuse timers in the network. However, in our simulations, N_h stays at 5. The reason is that route suppression keeps the intensity of path exploration at an almost constant level. The heavier path exploration is, the sooner route suppression is triggered, and the more potential messages are eliminated. This is what damping was designed for. Table IV lists the average number of updates each node receives from one peer during the charging period. The first seven topologies are mesh topologies with different sizes and node degrees, and the last two are Internet-like topologies with long-tailed distribution of node degree. In average, a node expects to receive the similar number (6 - 7) of updates from each link even though the network size and node degree increase significantly. Therefore, although N_h depends on flapping interval (Fig. 10 (c)) and damping parameters (Fig. 12), it is *insensitive* to the increase of network size and node degree.

V. Related Work

Though route flap damping is an important mechanism in BGP to deal with route instability, it has not received much attention from academic research. RFC 2439 [13] describes its design rationale, algorithm and implementation strategy. RIPE-229 [10] publishes recommended parameter settings and calls for "progressive damping". RFC 3221 [3] mentions that route flap damping is deployed widely, but there is also evidence of inconsistent parameters and partial deployment. None of these documents have discussed the reuse timer interaction and routing dynamics introduced by damping in the Internet. Mao et. al. [8] shows that one pulse is able to trigger route suppression even in a small topology due to path exploration. This finding is also supported by experiments on the Internet using BGP beacon [9]. Lad et. al. [7] analyze BGP trace data during worm attack and find damping may not be deployed by some routers.

Labovitz et. al. [5][4] observed BGP slow convergence in the Internet. Their analysis shows that during route changes, BGP can potentially explore a large number of alternate paths, resulting in excessive updates and long convergence time. BGP-Assertion [11] attempted to reduce path exploration by enforcing some route sanity checks. Ghost Flushing [2] let router withdraw a route first, while the route announcement is held by the MRAI timer. This approach reduces convergence time when a prefix is unreachable, but its extra withdrawals may interact with damping.

VI. Summary

In this paper we used simulation tools to examine the impact of damping on the BGP routing dynamics. We find that route suppression and reuse at one router can affect the number of routing updates received by other routers, and in turn, others' damping behavior. We explained how this interaction between route reuse timers at different routers, when compounded with BGP path exploration, can lead to a staged behavior of routing update propagation. Our results show that damping can confine global routing dynamics to follow a predictable analytical model when connectivity to a destination flaps persistently. However when the number of flaps is small, the global routing behavior deviates from the intended analytical model and damping leads to higher dynamics as measured by both message overhead and network convergence. This basic pattern of such damping-induced routing dynamics occurs with a wide range of flapping interval, damping parameters, and network topology. The damping parameter settings can have significant impact on the convergence delay and message overhead, both of which can be reduced by speeding up route reuse.

Although our results are obtained from simulations with a number of simplifications, such as modeling BGP as a simple path-vector protocol with shortest-path first policy and each AS as a single node in the topology, and examining damping behavior with a constant interval of route flapping, we believe that the the reuse timer interaction, as exhibited in the simulation, is likely occurring in the real Internet. First, BGP slow convergence and path exploration have been observed in the Internet [5], and BGP path exploration indeed triggers route suppression [9]. Second, when multiple BGP routers set reuse timers, if any of these reuse timers blocks the announcement of the preferred path to a flapping destination, it will have a noisy expiration, the sufficient condition for one reuse timer to affect another. Our simulation confirms that this can happen in Internet-like topologies. Third, our simulation assumed an ideal setting of full damping deployment with consistent parameter settings for the whole network, while in the real Internet it is known that damping is neither uniformly deployed nor damping parameters set consistently. With inconsistent damping parameters, reuse timer interaction can happen even without path exploration. For instance, in a single-line topology A-B-C, A is the flapping source, and C punishes route flapping more aggressively than B does. After A flaps several times, by the time B reuses its route to A, C's reuse timer is still on, and will be postponed by the announcement from B.

Our work serves as yet another illustration that large-scale distributed systems, such as the Internet, tend to behave in various unexpected ways. A design that works well in small scale may not function as intended in a large scale system. One of the causes for unexpected behavior is unforeseen feature interactions, as exemplified by the interactions between damping and BGP's path exploration, and between different reuse timers. It may be unrealistic to expect a new protocol design to foresee and eliminate all the potential interactions beforehand. Close monitoring of running systems and re-examinations of deployed protocol mechanisms are necessary, so that such detrimental interactions can be identified and remedy applied.

This work is the first step towards fully understanding the effect and effectiveness of damping mechanism. Despite its unintended behavior under certain conditions, we firmly believe that damping is a necessary mechanism to protect the global Internet routing infrastructure from melting down under high routing dynamics. Although each specific problems can be fixed, nevertheless damping serves as the last fence against overloading when other mechanisms have failed.

Our future work includes studying the impact of damping on data packet delivery, the behavior of partial deployment, and solutions to eliminate false triggering of route damping due to unexpected routing dynamics such as path exploration.

REFERENCES

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

[2] Anat Bremler-Barr, Yehuda Afek, and Shemer Schwarz. Improved bgp convergence via ghost flushing. In *Proc. of IEEE INFOCOM*, 2003.

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

[4] C. Labovitz, R. Malan, and F. Jahanian. Origins of internet routing instability. In *Proc. of IEEE INFOCOM*, 1999.

[5] Craig Labovitz, Abha Ahuja, Abhijit Abose, and Farnam Jahanian. Delayed internet routing convergence. In *Proc. of ACM SIGCOMM*, 2002.

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

[7] Mohit Lad, Xiaoliang Zhao, Beichuan Zhang, Dan Massey, and Lixia Zhang. An analysis of bgp update burst during slammer attack. In *Proceedings of the 5th International Workshop on Distributed Computing*, 2003.

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

[9] Z. Morley Mao, Randy Bush, Tim Griffin, and Matt Roughan. Bgp beacons. In *ACM SIGCOMM Internet Measurement Conference (IMC)*, 2003.

[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, X. Zhao, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang. Improving bgp convergence through consistency assertions. In *Proc. of IEEE INFOCOM*, 2002.

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

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