An Empirical Study of Router Response to Large BGP Routing Table Load

Di-Fa Chang Ramesh Govindan John Heidemann

Abstract— **Anecdotal evidence suggests that misconfiguration of backbone routers occasionally leads to an injection of large routing tables into the BGP routing system. In this paper, we investigate the detailed mechanics of router response to large BGP routing tables. We examine three commercial routers, and find that their responses vary significantly. Some routers exhibit *table-size oscillations* that have the potential to cause *cascading failure*. Others need operator intervention to recover from large routing tables. We also find that deployed resource control mechanisms, such as prefix limits and route flap damping, are only partially successful in mitigating the impact of large routing tables.**

I. Introduction

The behavior of modern-day routers under stress has not received much attention in the research literature. A particular aspect of router behavior under stress is the response of routers to *injection of large routing tables*. The network operator community closely monitors global routing table sizes, and most backbone routers are, nowadays, usually configured with several times more memory than that require to support the prevailing global backbone routing table size. However, in the past, *router misconfigurations* at a single router have been known to inject, for brief periods of time, very large routing tables into the routing system. Little is known about the mechanics of router behavior under these circumstances. The contribution of this paper is to present a measurement technique to evaluate this case, observe the behaviors of several commercial routers under stress, and evaluate how current router resource control mechanisms interact with load.

We monitor the mechanics of router behavior with two kinds of route loading experiments:

- We set up BGP connections with a commercial router and microscopically observe the effects of loading the router with a routing table near or over capacity. This experiment provides the understanding of the detailed mechanics on the routers under stress.
- We examine the behavior of simple BGP topologies under large routing table loads to understand how overloading a single router can impact other routers in the topology.

We find several interesting results:

- Some commercial routers exhibit a "malloc failure" when confronted with a large routing table from one or more peers. Such a failure essentially causes a soft reset of all router state, after which the router proceeds to re-establish the BGP peering sessions, and then fails again. This sequence can, in the absence of manual configuration, repeat itself indefinitely. During this sequence, the routing table size of the router repeatedly *oscillates* between zero and the maximum permitted by the router's capacity.
- When a chain of routers is configured so that each router in the chain peers with a neighboring router, this routing table-size oscillation can propagate down the chain. Thus, routers in the topology are subjected to routing table *flaps* (changes to route entries). In some cases, a "malloc failure" induced in one of the routers can actually *cascade* to the others and such a cascade may *persist*. This phenomenon is dependent on the relative memory configurations of the routers in the chain and on the speed of route processing at each of these routers.
- Other router implementations disable interfaces, or automatically disable BGP peering with routers from whom they received large routing tables. Operator intervention is required before peering can be established.
- For all the routers we studied we found that large-routing table loads do not cause packet traffic forwarding delays or drops except when the router flushes the entire forwarding table during a reset.

There exist at least two mechanisms (*prefix limiting* and *route-flap damping*) in modern routers that give the operator some control over router resource usage. We also investigate whether these mechanisms help alleviate some of the adverse effects of large routing table loads.

To our knowledge, no empirical study of router response to large BGP routing table loads exists in the research literature. In an extended version of this paper [4], we discuss tangentially related work. Here, we briefly discuss an ongoing standardization effort to add a "graceful restart" capability to BGP [5]. This work suggests keeping forwarding state across TCP resets. It also proposes to use an empty UPDATE message as an End-of-RIB marker that indicates to its peer the completion of the initial routing update after the session is established. The restarting router defers route selection until it receives the End-of-RIB markers from all peers. These mechanisms allow the router to continue forwarding traffic while it re-establishes BGP peering sessions, a functionality called "graceful restart." This mechanism is somewhat complementary to the impact of large routing loads that we consider. It can possibly preserve routing forwarding capability across some of the kinds of failures we consider.

II. Experimental Setup

We conduct our experiments on two canonical topologies: a *single* router topology (Fig. 1), and a *multiple router* topology (Fig. 2). In the figures, we denote the advertising BGP speak-

The authors are with USC/Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, CA 90292, USA. (email: {difac, govindan, johnh}@isi.edu) This work is partially supported by DARPA via the Space and Naval Warfare Systems Center San Diego under Contract No. N66001-00-C-8066, and by NSF grant number ANI-9986208.

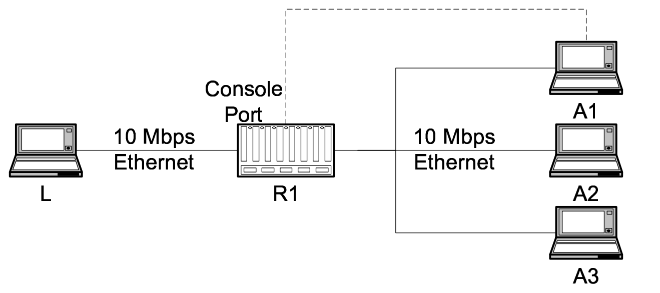

Fig. 1. Network topology used in single router experiments

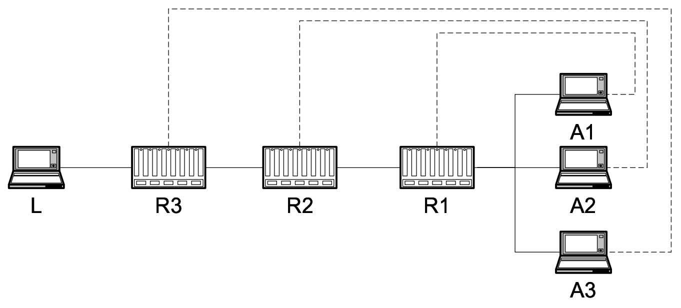

Fig. 2. Network topology used in simple topology experiments

ers by A_i, the routers under test by R_i, and the observing BGP speaker by L. The advertisers are responsible for injecting BGP routing tables to the test router, while the observer simply peers with the test router without advertising any route. Both advertisers and observer log their interaction with the test router for later analysis. The three commercial routers under test are shown in Table I.

To examine the router's response under different patterns of routing load, multiple BGP speakers (three advertisers and one observer) are used to peer with the test router such that each peer can generate different routing loads. Our advertisers and observer are FreeBSD machines running GateD 3.6. This choice is partly dictated by our lack of access to many commercial routers, but it helps by providing flexible logging facilities. As indicated in many router testing reports, turning on a commercial router's logging facilities (typically syslog to a remote UNIX machine) can dramatically degrade the router's performance. Using host-based BGP peers provides a light-weight alternative to record the router's external behavior.

To monitor the internal state of the routers under stress, a separate line is connected to the router's console. We develop a program that can continuously execute commands on the router's console and record the output, *e.g.*, the command `show ip route summary` on Cisco's router displays the statistics of the current routing table. The execution of these commands places less burden on the router performance than the debug and logging facilities provided by the router OS. However, heavy processor load can delay execution of these commands. We observe delays of up to 10 seconds during router failure.

Each advertiser generates some large number of routes, depending on the experiment. All routes in our experiments are prefixes of length 24 bits. Different advertisers generate distinct prefixes. For all experiments described in this paper, the routers

TABLE I. Configuration of Tested Routers

Router	Memory	OS version
Cisco 7000	64MB	IOS 11.1
Cisco 12008 GSR	64MB	IOS 12.0
Juniper M20	768MB	JUNOS 4.3
Juniper M20	768MB	JUNOS 4.4

have no policy configuration for route import and export. In other words, all distinct sets of routes from different advertisers will be installed into all routing databases in the router.

To better understand how various routing events relate to each other, we use NTP (Network Time Protocol) to synchronize all machines' clocks. All our logging facilities (GateD logging and router console logging) record the time associated with the routing events including the sending and receiving of routing messages, router failures, routing table changes, etc. In this way, we can generate the timing figures for each routing event. Overlapping these figures by aligning the timestamps indicates the relationship between routing events.

III. Router Response to Large Routing Tables

In this section, we present the results of experiments (Fig. 1) that load a single router with a BGP table exceeding the router's memory capacity. These experiments are designed to explore the failure modes of a single router in isolation and to serve as a basis for multi-router studies. In this and the next section, we describe router behavior using a simple memory model for BGP routers: the router has a BGP database that stores routes learned from or advertised to neighbors, a RIB that contains the selected routes, and one or more FIBs installed in router line cards.

A. Cisco 7000 with IOS 11.1

Our Cisco 7000 [1] (IOS 11.1) is capable of storing about 130,000 BGP routes. Initially, A_1 and A_2 announce 70,000 and 50,000 routes, respectively. After these 120,000 routes have been processed in the router, A_3 announces 20,000 routes to the router. This last set of advertisements exceeds the router's memory capacity.

Fig. 3 shows the number of routes stored in the router's databases as time proceeds. Our monitoring program extracts this information from the console of R_1. Note that the BGP database grows stepwise instead of continuously, and waits for routing table to catch up before growing. This lag represents the time taken to process the routes (apply policies, insert into the tree structure, install routes into the line card).

Now we focus on the moment of overloading. At time 1306, A_3 starts announcing routes and R_1's BGP database size increases accordingly. When the BGP database size exceeds the memory capacity, R_1 declares a "malloc failure" and *all BGP peering sessions are closed*. Thus, even though R_1 has enough capacity to handle routes from each individual peer, and even though in our experiment it is A_3's advertisements that cause

[1] This deliberate choice of an older OS version allows us to study the possible impact of large table loads on older routers. We suspect not all ISPs run the latest versions of the OS. Furthermore, our choice allows us to understand earlier Internet meltdowns [4], [1].

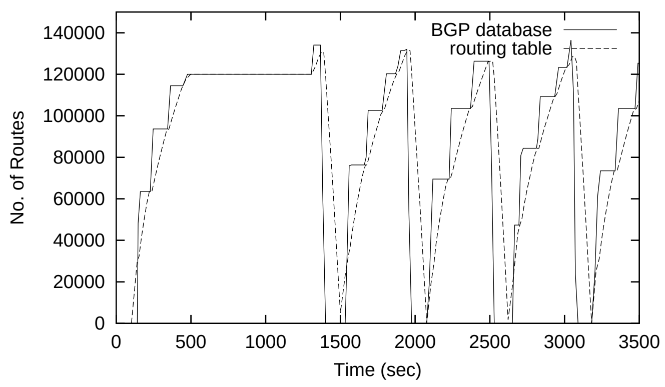

Fig. 3. Cisco 7000 under malloc failure.

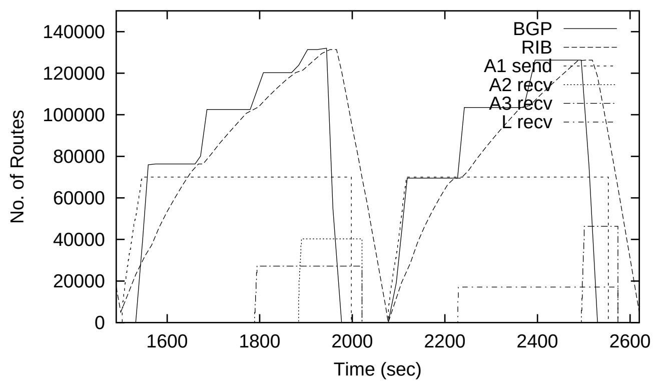

Fig. 4. Route installation and re-advertising in Cisco router with default MinRouteAdver (30 seconds).

R_1 to exceed its memory, it tears down the BGP sessions with A_1, A_2, and L as well. We observe that the router clears the dynamic routes in the routing table and resets all other routing processes including the intra-domain routing processes. Only static routes and directly connected networks (direct routes) survive in the routing table across the reset.

We note that the BGP specification itself does not specify what a router should do when it runs out of memory. However, the stateful design of BGP, in which a BGP speaker is required to remember routing information it learned from its peers, precludes a graceful degradation mode where the router can selectively "shed" excess routes. (A router can selectively "shed" peers; we shall see later that this is how other routers behave).

R_1 then attempts to *re-establish BGP peering sessions* with all its neighbors. The BGP specification does not mandate a particular behavior. This particular router implementation chooses to immediately reset all routing processes and re-establish BGP peering sessions. When the connections are re-established, all three advertisers start announcing routes almost simultaneously. Since the router has re-established connections to all peers, the load of 140,000 routes from A_1, A_2, and A_3 is processed and results in *another* malloc failure. This sequence can repeat indefinitely, causing a repetitive router failure as the entire routing table *oscillates*. Operator intervention is required to terminate this oscillation.

Now we turn our attention from the router's internal state to its external behavior, namely the sending and receiving of routing protocol messages. Fig. 4 shows how the route announcements propagate through the router during two up-down cycles of the router failure. Here we try to understand the temporal relationship between different routing events, namely, the advertisers announcing routes, the router importing the routes into BGP database, the router installing the routes into its routing table, and neighbors receiving routes re-advertised by the router. For ease of exposition, we show only A_1's announcements, but the result is the same for other advertisers' announcements. In the figure, "A_1 send" represents the cumulative number of routes announced by A_1 during the lifetime of each BGP session between it and the router. Similarly, "A_2 recv" represents the cumulative number of routes re-advertised by the router to A_2. "A_3 recv" and "L recv" are similarly defined. As in Fig. 3, the lines "BGP" and "RIB" represent the total numbers of routes in BGP database and routing table, respectively.

We study the number and the frequency of re-advertisement since this determines the impact on the router's neighbors. The result shows that the router re-advertises the routes only when the routing table is consistent with the BGP database. This complies with the BGP specification. As we indicated before, instead of processing the routes one by one, the router will import a fraction of received initial advertisements into BGP database, then gradually install them to the routing table. This may reflect the implementation decision of accelerating the route-selection process by batch execution.

As shown in the figure, the re-advertisements to A_2, A_3, and L do not happen in every failure cycle. In first cycle, only A_3 and A_2 get the re-advertisements, while in second cycle only L and A_3 receive the re-advertisements. We conducted other experiments by setting MinRouteAdver to 10 and 1 seconds, and observed that this does make the router re-advertise to all peers more frequently. Specifically, the router re-advertised to all peers 2 or 3 times in each failure cycle when the MinRouteAdver was less than 10 seconds. In Sec. IV, we illustrate the impact of these propagating routes on a larger topology.

B. Cisco GSR with IOS 12.0

With our Cisco GSR (ISO 12.0), our experiences are different. Since this router (with 64M memory) can store about 16,000 BGP routes, we have A_1, A_2, and A_3 announce 10,000, 5,000, and 2,000 routes, respectively. When receiving excess routing announcements, the router permanently stops responding to the interface where it peers with the advertisers. Thus, even though it is advertiser A_3's advertisements that causes the router to exceed its memory, the BGP sessions with A_1 and A_2 are also disconnected due to the interface failure. We omit the graph of this router behavior here, but it resembles the period from 0 to 1500 seconds of Fig. 3. Although the router is still able to process the console commands, we need to manually reboot the router to bring it back to normal operation. This different failure mode does not leak routes to its neighbors and has less impact on the infrastructure, in the sense that it does not exhibit table-size oscillations that can cause route flaps at neighboring routers. However, this mode *requires operator intervention* in order to restore connectivity; even if the advertising routers no longer announce large routing tables, R_1 has no way of re-establishing the BGP connection until the operator intervenes.

C. Juniper M20

When we repeat our experiment on a Juniper M20 with JUNOS 4.3, we find that the result is similar to that of the Cisco GSR. Unlike the Cisco routers, where a large routing table will overrun the BGP database before the routing table, our Juniper router has much more capacity in the BGP database than the forwarding table. As a consequence, a large routing table (about 500,000 routes for this router) overruns the forwarding table before the BGP database. The router completely stops responding to the interface where it peers with the advertisers. We need to unplug the cable from the interface card and re-plug the cable in order to resume the router's normal operation. We reported this bug to the vendor who then releases a new version of the OS with a fix.

After we upgraded to a new OS release (JUNOS 4.4), the router handled the failure more gracefully. When we send it the same number of routes, the forwarding engine generates a "malloc failure" error message indicating that some prefixes can not be installed into the forwarding table, but the interface remains operational. Thus, it appears that our experiment exceeds the Juniper's line card memory but not the route processor memory. Even though the FIB capacity is exceeded, the router continues to route packets for all prefixes. We are not sure how it continues to route packets; possibly it fills the line-card FIB on demand, or possibly it forwards packets not handled in the FIB through the RIB. Because we had access to the Juniper only for a very limited time we were not able load the router to the point where memory for the the BGP database or RIB would have been exhausted.

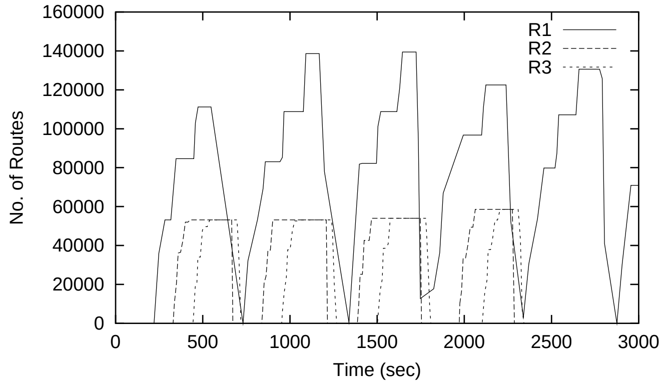

Fig. 5. Routers with same memory capacity

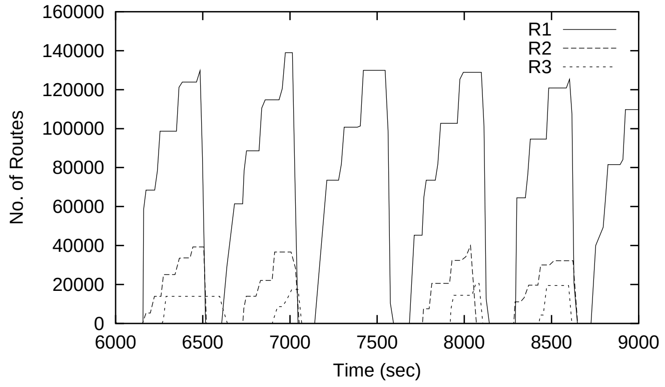

Fig. 6. Large-Medium-Small configuration

IV. Impact of Large Routing Tables on Simple Topologies

As described in Sec. III-A, injecting a large routing table into a single router can result in a table-size oscillation. A follow-on question is: how would table-size oscillation affect the network topology as a whole? We use a simple linear topology (Fig. 2) consisting of three Cisco 7000 routers, and vary the memory capacities of routers. This technique allows us to examine effects of router diversity on the impact of the failure, while at the same time allowing us to microscopically analyze router behavior. To emulate the effects of different physical memory capacities, we preload some of the routers with different numbers of static routes, and configuring the router not to re-advertise the static routes. By consuming some memory through software, less memory is left for BGP-learned routes.

A. Routers with the same capacity

In this experiment, all three routers are configured with the same amount of memory. This scenario is intended to study the impact of table-size oscillation in relatively homogeneous sections of the Internet topology, such as routers within a backbone network. Fig. 5 shows the changes in the BGP database size for the three routers when A_1 and A_2 overload R_1. (A_1 sends 80,000 routes at time 215 and A_2 sends 70,000 at time 274.) Before failing, the router (R_1) propagates some routes to R_2 which, in turn, propagates a subset of these to R_3. In this case, then, routers see their routing table fluctuate over time. R_1 periodically fails, and R_2 and R_3 see their routing tables change as R_1's routes are advertised and withdrawn over time. Thus, each router downstream of R_1 see table-size oscillations, albeit smaller in amplitude and not resulting in router failures.

As indicated in Sec. III-A, the number of propagated routes and the frequency of these fluctuations are determined by the speed of route processing in R_1 and R_1's MinRouteAdver timer for the BGP session to R_2. Also, we observe that sometimes routes are propagated down the chain of routers, but sometimes they are not. For example, the last peak in Fig. 5 (for 2300 to 2900 seconds) does not propagation routes. We believe that this failure to propagate routes is due to an interaction between R_1's batched advertising mechanism and its MinRouteAdver timer.

B. Large-Medium-Small

In this experiment, the routers are configured with decreasing memory capacities (R_1 is largest, R_2 is medium, and R_3 is smallest) and we inject a large routing table into R_1. This scenario is meant to mimic a misconfiguration in a large ISP, and its effects on customers and smaller downstream ISPs.

As in previous experiment, the failed router R_1 advertises some routes to medium router which then propagates to small router. Due to the batched advertising implementation in these routers, the number of routes advertised to R_2 is much less than the number of routes received by R_1. Depending on the memory configuration of the medium and small routers, we can observe an interesting phenomenon that we call a *cascading failure*. In

Fig. 6 we show an extreme experiment where R_1 is capable of dealing with 130,000 routes while R_2 and R_3 can only deal with 40,000 and 20,000 routes, respectively. We overload R_1 with 150,000 routes. Before R_1 fails, it advertises nearly 60,000 routes to R_2 which then also fails. Similarly, before R_2 fails, it propagates about 20,000 routes to R_3, sometimes causing R_3 to fail.

Note that in this case, powering down the medium or large router for some time *cannot alleviate* these failures because upstream routers or the advertisers will recreate the problem. That is, a downstream ISP cannot really eliminate the cascades by local actions such as power cycling equipment or installing filters (although installing filters can prevent the cascade from spreading downstream). The failure condition can only be eliminated by reconfiguring the advertisers to not exceed all routers' capacities.

V. Impact of Resource Control Mechanisms

The table-size oscillations described in Sec. III-A are caused by a resource overrun (in this case, memory). Two relevant resource control mechanisms are currently deployed in routers in the Internet: *prefix limiting* and *route-flap damping*.

A. Prefix Limiting

Prefix limiting is a mechanism that places a configured limit, called teardown threshold, on the number of prefixes that a router will accept from a given BGP peer. When the number of prefixes announced by a particular peer exceeds the teardown threshold, the router tears down the BGP peering session with its neighbor. Clearly, this feature prevents router memory overrun (and its attendant resetting of all BGP peerings (Sec. III-A)) caused by a single large routing table infusion from a peer. When a peer exceeds its prefix limit, only the router's BGP session to that peer is affected; all other peering sessions are kept alive, and the router continues to forward traffic to the unaffected prefixes.

To evaluate this, we repeat the experiment of Fig. 1 on our Juniper M20 running JUNOS 4.4 with prefix limiting. We set the teardown thresholds for all peers to 200,000. At time 250 A_1 announces 170,000 routes and at time 500 A_2 announces 40,000 routes. After the router R_1 processes these 210,000 routes, at time 1000, A_1 send routing updates consisting of additional 40,000 routes. Fig. 7 shows the result. At time 1000, the total number of routes announced by A_1 exceeds the configured limit, and so R_1 closes the peering session to A_1. A_1 re-establishes the peering session with R_1 and re-advertises the 210,000 routes. Again, R_1 tears down the BGP peering. Thus, this particular implementation of prefix limits *still exhibits table-size oscillations* that can only be stopped by operator intervention. In the experiment of Sec. III-A, however, the router resets all BGP peering sessions; here, only the session that causes the router to exceed its limits is torn down.

We conducted the same experiment on Cisco GSR with IOS 12.0, observing a somewhat different result. When R_1 detects an advertiser exceeding the threshold, it permanently denies further connection attempts from that advertiser. Thus, no table-size oscillations occur. However, operator intervention is required to

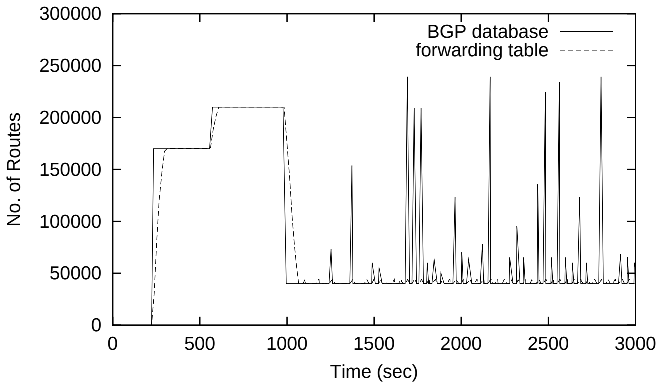

Fig. 7. Prefix limiting experiment on Juniper M20

re-establish the peering session, even if A_1 is no longer injecting a large routing table prefix.

B. Route Flap Damping

In Sec. IV-B we describe how a large-medium-small router configuration will propagate table-size oscillations and can result in cascading failures. *Route-flap damping* is a deployed mechanism in Internet routers that is used to limit the rate of propagation of unstable routing information [3].

We conduct experiments to investigate the effectiveness of this mechanism on damping the two types of route flaps described in previous sections: route flaps injected by a failed Cisco 7000 (Sec. III-A), and route flaps injected by a Juniper M20 using prefix-limiting (Sec. V-A). In all experiments, we use the vendor supplied default route flap damping parameters: a half-life of 15 minutes, a reuse threshold of 750, a suppress threshold of 2000, and a maximum suppress time of 60 minutes.

For the first type of route flaps, route flap damping cannot detect and suppress flaps. This is because in this case all BGP connections are terminated and the router loses all state, including the instability history associated with each route. Fig. 8 shows an example where we conduct the same experiment as in Sec. IV-A but enable route flap damping in all three routers with default damping parameters. In this case, as described in Sec. IV-A, R_1 does not propagate routes to R_2 periodically. Thus, in R_2, the instability values associated with the applied routes decay exponentially during the periods that no route readvertisements are received. When the period is long enough to have the instability values fall below the reuse threshold and R_1 propagates the routes again, R_2 reinstates these routes in its routing table (from time 4280 to time 4640 in Fig. 8).

We find that route flap damping can be effective with prefix-limiting. By using prefix-limiting, a router can keep instability history for prefixes advertised by the offending peer. When the peer repeatedly re-advertises its table, the router can suppress these routes. This is an apparently attractive reaction to the failure, in that it confines the impact of route flapping to the immediate peers of a router. Note however, that route flap damping is not without its drawbacks. Because of the way damping works, for several tens of minutes after the peer stops injecting a large routing table, the router continues to suppress these routes. This effectively ensures that the corresponding prefixes are unreach-

TABLE II. Summary of results of experiments without prefix limiting

Router	Table Overloaded	Response	Impact	Peers Affected	Damping Efficacy
7000 (IOS 11.1)	BGP database	reset BGP process	repeated session resets	all	sometimes
GSR (IOS 12.0)	BGP database	freeze interface	link failure	all peers on affected interface	N/A
M20 (JUNOS 4.3)	forwarding table	freeze interface	link failure	all peers on affected interface	N/A
M20 (JUNOS 4.4)	forwarding table	possibly degrade forwarding performance	low forwarding performance	no	N/A

TABLE III. Summary of results of experiments with prefix limiting

Router	Prefix-limiting Efficacy	Prefix-limiting Action	Impact	Peers Affected	Damping Efficacy
GSR (IOS 12.0)	BGP database can still be overloaded	deny peering	no peering	one	N/A
M20 (JUNOS 4.4)	forwarding table can still be overloaded	reset peering	repeated session resets	all	all times

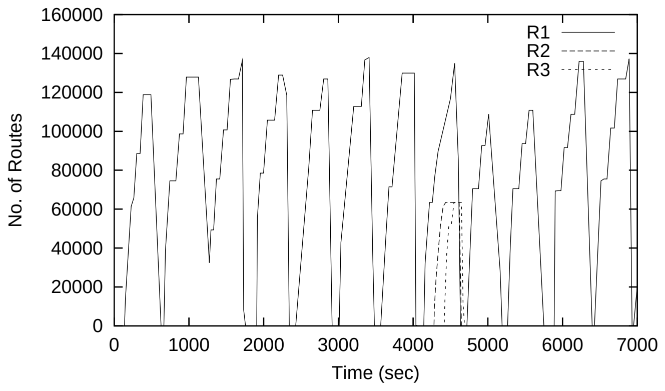

Fig. 8. Route flap damping experiment on three Cisco 7000's

able for extended periods of time.

VI. Conclusion

In this paper, we investigate the detailed mechanics of router response to large BGP routing tables, how the failure affects neighbors, and how well the resource control mechanism can alleviate the problem. Table II and III summarize our experimental results. We believe that BGP routers under other stressful conditions, *e.g.*DDoS attacks and worm spreading, can exhibit similar behavior. Thus, this study provides researchers an understanding on the failure handling capacity of routers in Internet.

References

[1] R. Barrett, S. V. Haar, and R. Whitestone, "Routing snafu snips net service," Apr. 1997, `http://www.zdnet.com/zdnn/content/inwk/0413/inwk0032.html`.

[2] Y. Rekhter and T. Li, *A Border Gateway Protocol 4 (BGP-4)*, RFC 1771, Mar. 1985.

[3] C. Villamizar, R. Chandra, and R. Govindan, *BGP Route Flap Damping*, RFC 2439, Nov. 1998.

[4] D.-F. Chang, R. Govindan, J. Heidemann, *An Empirical Study of Router Response to Large Routing Table Load*, Technical Report No. 552, USC/Information Sciences Institute, CA., Nov. 2001.

[5] S. Ramachandra, Y. Rekhter, R. Fernando, J. G. Scudder, and E. Chen, *Graceful Restart Mechanism for BGP*, `draft-ietf-idr-restart-05.txt`, Jun. 2002.