BGP Convergence Delay under Large-Scale Failures: Characterization and Solutions

Amit Sahoo, Krishna Kant, *Member, IEEE,* and Prasant Mohapatra, *Senior Member, IEEE*

***Abstract*— Border gateway protocol (BGP) is the default routing protocol between various autonomous systems (AS) in the Internet. In the event of a failure, BGP may repeatedly withdraw routes to some destinations and advertise new ones until a stable state is reached. It has been found that the corresponding *convergence delay* could stretch into hundreds of seconds or more for isolated Internet outages and can lead to high packet drop rates. Previous studies on BGP failures have looked at isolated failures scenarios. In this paper[1] we characterize BGP recovery time under large-scale failure scenarios. We show that the recovery time depends on a variety of topological and BGP parameters, and can be substantial for massive failures. We also observe that the Minimum Route Advertisement Interval (MRAI) and the processing overhead at the routers during the re-convergence have a significant effect on the BGP recovery time. We propose two new schemes to bring down the processing overload at BGP routers, resulting in reduced convergence delays. We show that these schemes, combined with the tuning of the MRAI value, accelerate the BGP convergence process significantly, and can thus limit the impact of large scale failures in the Internet.**

***Index Terms*— Border Gateway Protocol (BGP), Large-Scale failures, Convergence Delay, Minimum Route Advertisement Interval (MRAI).**

I. Introduction

BGP (Border Gateway Protocol) [3], [4] is the predominant inter-domain routing protocol used in the Internet. BGP belongs to the class of *path vector* routing protocols, wherein each node advertises the "best" route for each destination to all of its neighbors. A BGP node stores all the paths sent by its neighbors but uses and advertises only the one that is "best" according to the policy in effect. When this primary path fails, BGP withdraws this path and selects the next best backup route. The new route is then advertised to its neighbors. However there is no guarantee that the backup route is still valid. In case the backup route has also failed, it will be withdrawn only after a withdrawal is sent by the neighbor which advertised it; and another backup route is chosen. This absence of information about the validity of a route can cause BGP to go through a number of backup routes before selecting a valid one. The cycle of withdraws/advertisements can continue for a considerable amount of time and this delay is known as the *convergence delay* (or *recovery time*).

Internet routing features other classes of routing protocols as well, such as the *link state* and *distance vector* protocols. However the flooding overhead of link state protocols makes them generally inappropriate for inter-AS use. Distance vector algorithms, on the other hand, suffer from the *count-to-infinity* problem, in which nodes may continuously increase their cost to reach an inaccessible destination. Therefore, distance vector and link state protocols are generally used within an AS and inter-AS routing primarily uses BGP because of its better scalability, flexibility and configurability. In particular, the scalability of BGP has been a critical facilitating factor in the explosive growth of the Internet over the last decade.

BGP is a critical part of the Internet infrastructure and hence there have been numerous studies [5], [6], [7], [8], [9], [10] to analyze the impact of BGP route changes. In particular, it was shown by Labovitz et al. [6] that the BGP convergence delay for isolated route withdrawals can be greater than 3 minutes in 30% of the cases and could be as high as 15 minutes. They also found that packet loss rate can increase by 30x and packet delay by 4x during recovery. There have also been efforts [6], [7], [9] to formulate analytical models for BGP convergence delay. These studies have identified factors that affect the convergence delay and computed lower and upper bounds. However the theoretical bounds for the time needed to remove routes to an unreachable prefix (T_{down}) are very large and tell us little about the actual delays. Furthermore, the bounds do not take the magnitude of the failure into account. So while the bounds are the same whether a failure involves one or a hundred routers, in reality the convergence delays do increase with the size of a failure (as long as the size of the failure is not too large). This points to the need for investigating the effects of large-scale failures.

Large-scale failures can be caused by a number of reasons such as earthquakes, major power outages, hurricanes, terrorist attacks, malicious attacks on the Internet infrastructure etc. While large-scale failures could be dispersed across the Internet, the majority of the causes would lead to a geographically contiguous area of failure and that is the default case that we analyze in this study. A "large-scale" affects multiple routers in the network, and typically spans multiple ASes. And as part of this study we simulated a wide range (in terms of magnitude) of large-scale failures.

Besides significantly degrading the connectivity from and to the affected ASes, large scale failures will also have a big impact on the connectivity between the source-destination pairs that use the affected ASes for transit. Rexford et al. [11] observed that the routes to the most popular prefixes/ASes in the Internet are remarkably stable. They conjectured that this was because the network equipment in those domains was well maintained. However a large scale failure far away

Amit Sahoo and Prasant Mohapatra are with the Dept. of Computer Science, UC Davis, CA 95616. Email: {asahoo,pmohapatra}@ucdavis.edu

Krishna Kant is with Intel Corporation, Hillsboro, OR 97124. Email: krishna.kant@intel.com

[1]Preliminary versions of parts of this work were presented at ICC 2006 [1] and DSN 2006 [2]

from these popular prefixes/ASes can still tear down routes to these destinations from large parts of the Internet. Furthermore communication networks are needed the most during times of crisis, and that increases the importance of a quick recovery. Therefore a quick recovery is crucial even for a large scale failure.

Our research efforts are focused on the recovery characteristics of BGP networks after large-scale failures and the factors that affect the convergence process. Towards this purpose we engineer failures of different magnitudes (0.25-10% of all routers) in a simulated network. While we do realize that 10 or even 5% failures in the whole Internet are highly unlikely, the failure magnitudes make much more sense if the network inside a geographical region (e.g. a small country or a state in the US) is being considered. Moreover, if we consider the actual number of routers that are failed (our networks contain 450-1000 routers), then the failures are definitely realistic. The details of our simulations are presented in Section III. We use the convergence delay as the metric to study the BGP recovery process after a failure. Once again, while the convergence delay might not be important if a 10% failure in the Internet takes place, it is of interest if we consider regional networks, or if we consider the actual number of failed routers. The convergence delay can be thought to be the period of routing instability in the network, and therefore from the perspective of a routing protocol like BGP, the goal should be to minimize this delay.

In this study we analyze the relative impacts of the size of failures, topological characteristics, the update processing overheads, and Minimum Route Advertisement Interval (MRAI) [3] on the convergence delay. Based on the quantitative studies, we propose a scheme to dynamically select the MRAI so that the rate of generation of update messages during large scale failures can be controlled. We also propose a novel batching scheme that reduces the number of route advertisements during periods of instability by suppressing the effect of update messages that are stale or redundant. We show that both the dynamic and the batching scheme can substantially reduce the convergence delays.

The rest of the paper is organized as follows. We discuss the previous work on BGP convergence delay in Section II. Section III outlines the details about the tools and the configurations that we used for our experiments. In Section IV we analyze the behavior of BGP convergence delay for large-scale failures. We present and evaluate our schemes to reduce the convergence delay in Section V. We summarize the results and overview future work in Section VI.

II. Related Work

There has been a fair amount of work on the analysis of BGP convergence properties and many parameters affecting the convergence time have been identified. However, most publications have examined simple networks or a specific set of sources and destinations only. In this section we talk about the important papers published in this area and the conclusions therein.

Previous works [6], [7] have concluded that the Minimum Route Advertisement Interval (MRAI) [3], [4] is one of the most important BGP configuration parameters affecting the convergence delay. The MRAI governs the rate at which a BGP router can send route advertisements to a neighbor. After a router has sent an advertisement to a neighbor, it has to wait for at least the MRAI before it can send a new route advertisement for the *same* destination to the *same* neighbor. The straightforward way to implement the MRAI would be on a per-destination basis, i.e. maintain a separate timer for each destination and each neighbor. The timer is started when the router sends an update for the corresponding destination to the neighbor in question. Thus, the next update can be sent only after the timer has expired. However the large number of destinations in the Internet makes this approach unviable and a per-peer scheme is more prevalent in the Internet today. In the per-peer scheme, the router maintains just one timer per neighbor and that timer is used to control the updates for all the destinations. This approach makes the scheme more scalable.

Labovitz et al. [6] developed a model for BGP convergence and showed that the convergence delay after a route withdrawal in a complete graph with *n* BGP nodes is *(n-3)*MRAI* at best and *O(n!)* at worst. They [12] later extended their model and determined that the upper bound for the time required for a route to converge is dependent on the MRAI and the length of the longest path from the source to the destination. Pei et al. [7] developed a more general model in which they also considered the processing delay for an update message. They considered scenarios where the BGP nodes were not overloaded and derived upper bounds for the convergence delay for such scenarios. However as we mentioned in the previous section, the derived bounds for T_{down} convergence delay are very large and the models do not take the size of the failure into account.

Griffin and Premore [8] studied the effect of MRAI on the convergence delay after a fault in simple BGP networks. They found that as the MRAI is increased, the convergence time first goes down to a minimum and then increases linearly. They observed that the optimal MRAI was dependent on the size of the network, the configured processing delay for the update messages, and the path-vector scheme in use. In particular, they found that the optimal value increased with an increase in the processing delay and the network size. The authors also looked at the variation in the number of update messages as the MRAI was increased and found that the message count decreased until the MRAI was close to the optimal value and then remained constant. The authors concluded that the default value of 30 seconds for the MRAI is "somewhat arbitrary" and in the ideal scenario we would have a different MRAI for each AS.

There have been a number of proposals to improve the BGP convergence delay after failures or changes in the network. We compare our schemes against two of the schemes that have been cited the most, Ghost Flushing [13] and Consistency Assertions [14]. Ghost Flushing proposes to improve BGP convergence by removing invalid routes (ghosts) quickly from the network. In normal BGP, a route advertisement might be delayed because only one route (for a particular destination) can be sent to a neighbor in one Minimum Route Advertise-

ment Interval (MRAI). Note that the route advertisement not only advertises a new route but also serves the purpose of withdrawing the older, possibly invalid, route. Therefore a delay in sending out a new route could cause the neighbor to use an invalid route for a longer period of time. To make matters worse, the neighbor could also forward the invalid route to other nodes. Ghost Flushing solves this problem by sending out an explicit withdrawal without waiting for the MRAI timer to expire, if the new route has a lower degree of preference than the older route. Consistency Assertions tries to identify and remove invalid routes from the routing tables. The basic idea is that if a path advertised by one neighboring AS (A) contains another neighboring AS (B), then the paths (to the corresponding destination) advertised by both the neighbors must be consistent. If they are not, then the directly learnt route (from B) is considered to me more dependable than the indirectly learnt route (from A), and the route from A is marked as "infeasible". Similarly, if the route from A contains B, but B has not advertised a route to the corresponding destination, the route is considered infeasible.

As our BGP modification schemes are closely linked to the MRAI parameter we should mention that Deshpande and Sikdar [15] proposed two MRAI related methods to reduce the convergence delay. The first method cancels a running MRAI timer if that can improve the convergence delay and the second method uses the MRAI for a destination only if the route for that destination has changed at least a specified number of times. The authors showed that these schemes reduced the convergence delay; however the number of update messages went up considerably. The Differentiated Processing Scheme proposed by Sun et al. [16] shares some similarities with our Batching scheme, as it is also designed to change the order in which updates are processed. However the criteria used to reorder the updates are different. Our scheme is more light weight and primarily designed to reduce the number of updates and the processing load at the routers.

III. Methodology

We used a number of synthesized topologies for our studies and varied their parameters to analyze the effect of these parameters on the recovery times. A modified version of BRITE [17] was used for topology generation and BGP simulations were carried out using SSFNet [18].

A. Topology Generation

BRITE can generate topologies with a configurable number of ASes and with multiple routers in each AS. BRITE supports a number of AS topology generation schemes such as Waxman [19], Albert-Barabasi [20], and GLP [21]. In the Waxman scheme, the probability of two ASes being connected is proportional to the negative exponential function of the distance between the two ASes. The Albert-Barabasi and GLP models try to generate a power-law degree distribution. However, the results are generally not satisfactory if the number of nodes (ASes) is less than a thousand. In order to rectify this problem, we modified BRITE so that it accepted a degree distribution as input and generated the interconnections according to this distribution. This provided us with complete freedom as far as degree distribution is concerned, and allowed us to experiment with distributions with different decay characteristics and distributions extracted from real networks besides uniform and constant degree distributions. We also modified the code to generate variable number of routers for the ASes. The number of routers in each AS was generated using a heavy tailed distribution and was in the range [1..100].

Geographical placement is essential for studying large scale failures since such failures are mostly expected to be geographically contiguous (e.g. an earthquake zone). However, directly using the geography of actual Internet is not only difficult (precise identification & location of routers is a hard problem) but also considerably limits the scenarios that can be studied. Instead, we placed all ASes and their routers on a 1000x1000 grid. Studies of real internet have found that the geographical extent of an AS is strongly correlated to the AS size (i.e., number of routers in the AS) [22]. Here we assume a perfect correlation and make the geographical area (the region over which the routers of an AS are placed) of an AS proportional to its size (number of routers). In particular, the routers of the largest AS are distributed over the entire grid. For smaller ASes, the area is reduced proportionately. The routers of an AS are distributed randomly over the geographical area assigned to it.

Internet studies also show that larger ASes are better connected [23]. This is handled as follows: We first create a sequence of AS degree values according to the selected AS degree distribution and sort them. Similarly, the AS list is also sorted according to the number of routers in the ASes . The degree of an AS is then set to the value at the corresponding location in the inter-AS degree list. This creates a perfect correlation between AS sizes and degree. Again, although a perfect correlation is unlikely in practice, it is a reasonable approximation for our study.

Although we normally did not take geographical location into account when creating inter-AS edges, we did run a few cases where we used a Waxman (distance-based) connectivity function. The ASes are connected together using a pseudo-preferential connectivity model in which one of the ends of the edge is selected randomly but the other end is selected according to the degree of the AS. Once the two ASes for an inter-AS edge have been determined, we randomly select a router from one of the ASes and preferentially connect it to a nearby router in the other AS. We used the default Waxman scheme (in BRITE) for creating the intra-AS edges. However we observed that distance based connections inside the ASes did not have any significant impact on the convergence delays. For all links, we used a one way delay of 2.5 ms (cumulative transmission, propagation and reception delay).

For most of our experiments we used 120 AS topologies. This was dictated partly by the fact that the Java Virtual Machine could allocate a maximum of 1.5 GB of memory on the 32 bit machines that we used and hence we could simulate at most $\sim$250 ASes . The benefit of using the 120 AS topologies was that we could verify the results using networks that were half as big (without being really small) and twice as big (still within the scope of our experimental

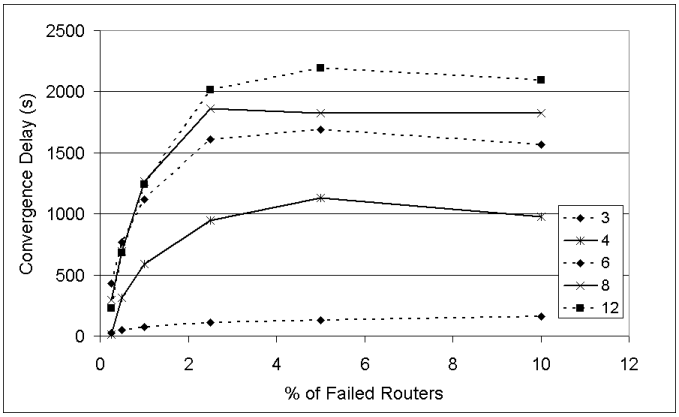


Fig. 1. Recovery time for constant degree networks

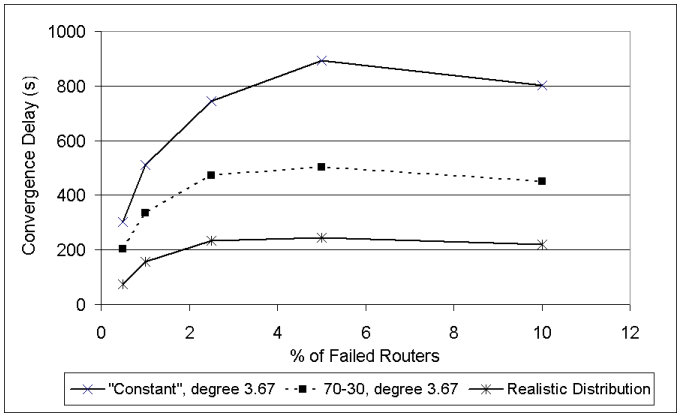


Fig. 2. Recovery time for different degree distributions

setup). The running time for the BGP simulations with 120 AS networks was also much more manageable than 200 or 250 AS networks, and this allowed us to experiment with many more scenarios and schemes. We generated 20 random topologies for each degree distribution that we experimented with. We simulated the failures on each of these topologies and averaged the values for convergence delays and number of messages. As we mentioned earlier each AS in the network had 1-100 routers. For the 120 AS topologies, the number of routers ranged from about 450 to about 1000.

B. BGP Simulation

We used the SSFNet simulator for our experiments because it has been used extensively in the research community for large scale BGP simulations and BRITE can export topologies in the format used by SSFNet. We used OSPFv2 as the intra-domain routing protocol. For BGP, *path length* (i.e., number of ASes along the route) was the only criterion used for selecting the routes and there were no policy based restrictions on route advertisements. All the timers were jittered as specified in RFC 4271 [3] resulting in a reduction of up to 25% in the timer period. In our experiments the MRAI timer was applied on a per-peer basis rather than a per-destination basis, as is commonly done in the Internet. We experimented with different eBGP (BGP connection between two routers from different ASes) MRAI values and discussed in Section IV-B. We used a mesh of iBGP (BGP connection between two routers from the same AS) peering instead of route reflection [3] inside the ASes as the number of routers in the ASes is not very large. The iBGP MRAI was always set to 0. The BGP update processing delay was modeled using the mechanisms available in SSFNet. We simulated failures by making all the routers in an area inoperative at the same time. After the routers were disabled, we studied the BGP recovery process by measuring the convergence delays and the number of generated messages.

IV. Characterization of BGP Convergence Delay

In studying the impact of large-scale BGP router failures on recovery time, the following parameters are the most relevant:

1) Magnitude of failure, in terms of the number of routers.
2) Inter-AS degree distribution (average degree & its variability).
3) MRAI value.

We study the effects of these and some other factors through our experiments. Our initial experiments indicated a considerable variability and complexity in BGP recovery time behavior. Consequently, for the experiments discussed below, we varied only one parameter at a time and also considered several simple topologies in addition to those modeled after real topologies. As mentioned earlier we experimented with a variety of MRAI values for our experiments. But, unless the MRAI(s) is/are explicitly specified, the results were obtained with a default eBGP MRAI of 30 seconds.

A. Degree Distribution

We first examined the variation in the convergence delay as a function of the average inter-AS degree. We started off with topologies in which all the ASes have a constant inter-AS degree. To avoid contamination of results due to other factors, distance wasn't considered while creating the inter-AS edges. Fig. 1 shows the recovery time (in seconds) as a function of the failure magnitude (in terms of fraction of routers failed). In all cases, the recovery time increases initially with the size of the failure to some maximum value and then slowly rolls off. The recovery time rises initially because, a larger failure translates into more failed routes and more failed backup routes. However as the number of failed routers continues to grow, the residual network gets smaller and hence the length of the backup routes explored during the convergence process is shortened. This causes the eventual decline in the convergence delay. It must be noted that the loss in connectivity in the network must keep increasing with the size of the failure. However, we are only looking at the BGP convergence delay here. It is also seen that a higher degree consistently increases the recovery time. This happens because the number and the lengths of possible backup paths goes up as the degree is increased.

We then investigated how the recovery time for a network with a "realistic" degree distribution would compare against that for a network with constant or uniform degree. For this purpose we decided to use the actual inter-AS degree

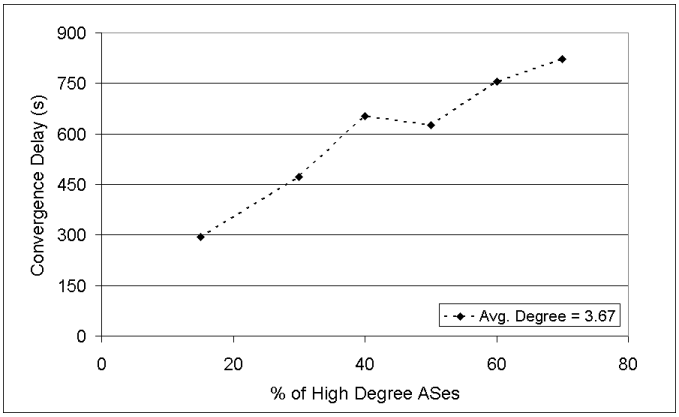


Fig. 3. Recovery time vs. percentage of high degree ASes

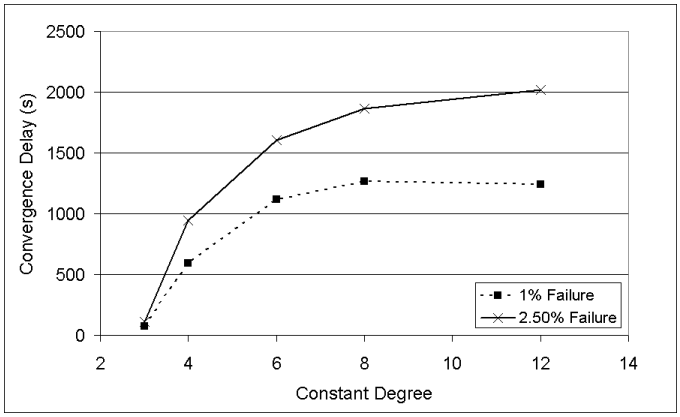


Fig. 4. Recovery time vs. constant degree

distribution in the Internet. The average measured inter-AS degree from the Internet AS-level topology is about 8.0 [24]. However, the Internet has over 22000 ASes and the maximum inter-AS degree is in the thousands. Therefore, we used the degree distribution derived from the Internet AS-level topology but decided to restrict the maximum degree to 40 for our 120 AS network . This gave us a degree distribution which decays as a power law with an exponent of about -1.9. The average degree is about 3.67. When we experimented with a topology that had a near constant inter-AS degree distribution (degree was either 3 or 4) with an average degree of 3.67, we found that this topology had convergence delays 3-4 times as high as the realistic case. This prompted us to closely examine the recovery time as a function of the degree distribution.

Fig 2 compares the convergence delays for the realistic topology mentioned earlier, a topology with "near" constant inter-AS degree of 3.67 and a third topology (referred to as the 70-30 case henceforth) where 70% of the ASes have low connectivity (1-3) and the other 30% have a high connectivity value (7 or 8) such that the average degree is again 3.67. It is seen that variable connectivity helps bring down the maximum recovery time considerably. Thus, the average degree is not a reliable indication of the recovery time. The reason for this behavior is that the overall recovery time is a result of two factors with respect to degree:

A Number of routes: Higher degree translates into more routes, which means that during a failure, the number of withdrawn routes as well as backup routes is higher.

B Route Lengths: Higher degree ASes however reduce the distance between other ASes. This leads to a shorter average backup path length and quicker propagation of updates. In other words, high degree ASes can act as "short circuits" and actually help lower the recovery time. It must be noted though that an increase in the average degree (while keeping the distribution the same) will increase the length of the longest paths in the network.

Thus a uniform increase in the degree of most ASes results in higher recovery times as shown in Fig 1. However increasing the degree of some ASes while keeping the average degree the same will do the opposite. This can be seen in Fig 2 where the convergence delay for the 70-30 case is less than the topology with constant inter-AS degree. Thus, the presence of a small percentage of high degree ASes can provide the beneficial short circuit effect and lower the recovery time. This can be seen more clearly in Fig 3 which shows the maximum recovery time as a function of the fraction of ASes that have a high degree. Recall that in the 70-30 distribution, 30% of ASes have a high degree (7 or 8) and the rest have lower degree (1-3). In Fig 3, we use a similar idea except that percentage of ASes with high degree is varied while maintaining the same average degree. As the fraction of high degree ASes decreases, their degree goes up. Fig. 3 shows the curve for average degree equal to 3.67. It is seen that the curves show a definite increasing trend. This reinforces the idea that a small number of well connected ASes among a large number of poorly connected ASes forms the ideal situation for low recovery time.

The arguments above still fail to explain why a distribution (e.g., power law) should yield lower recovery time than the fixed low-high mixture of degrees. This result follows by applying the above arguments recursively. We can lower the recovery time by again splitting the high degree fraction into parts: a large subset with lower than average degree, and a smaller subset with a much higher degree. Note that a recursive high-low degree partitioning is akin to cascade multifractal construction and in the limit yields the log-normal distribution.

One issue that we have not looked at is the behavior of the convergence delay as a function of average degree (with the type of degree distribution being the same). This is shown more clearly in Fig. 4 where we show the convergence delay of 1 and 2.5% failure for topologies with constant inter-AS degree. It is seen that the curve shows a diminishing return behavior, which may appear counter to the explanation of effect (A) above. The explanation lies in the fact that the convergence delay depends on the lengths of the longest backup routes explored during the convergence process. If the degree is already high, increasing it further doesn't lead to a proportional increase in the lengths of the longest routes.

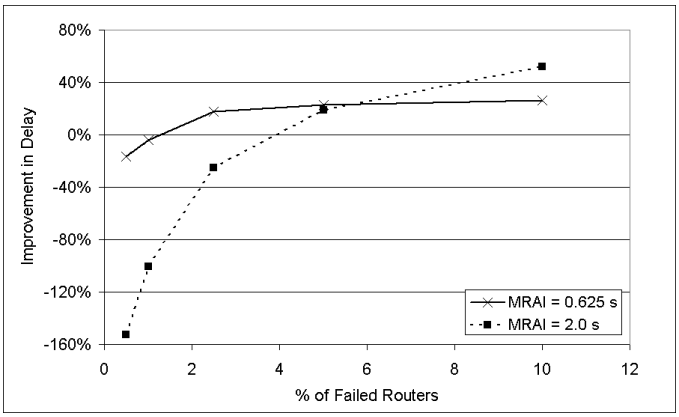


Fig. 5. Convergence delay for different MRAI values

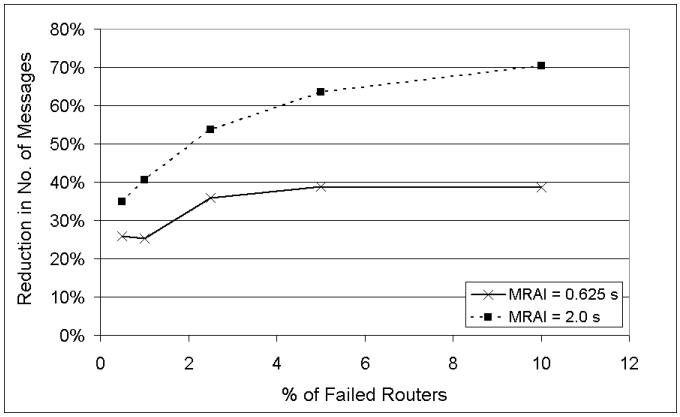


Fig. 6. Number of generated messages for different MRAI values

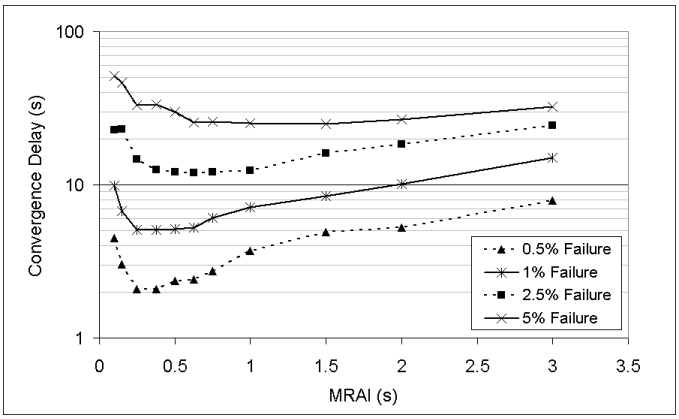


Fig. 7. Variation in convergence delay with MRAI

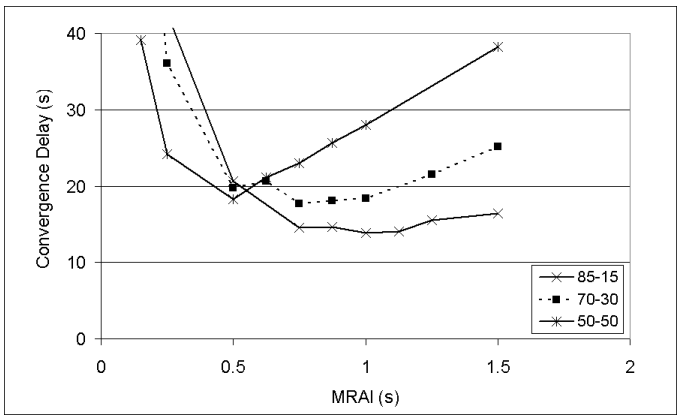


Fig. 8. Convergence delay for different topologies

B. Effect of MRAI

We have already seen that the convergence delay is dependent on the size of the failure. Now we investigate whether the MRAI value affects failures of different sizes differently. For this set of experiments we again used topologies with 120 ASes and the realistic degree distribution (average 3.67). We found that for these topologies, the "optimal" MRAI values (using which we get the lowest convergence delays) are much lower than the default value of 30 seconds, and we use these values for further experimentation. Fig 5 shows the relative variation in the convergence delay for different sized failures with three different MRAI values: 0.25, 0.625 and 2.0 seconds. In order to emphasize the effect of the MRAI, instead of plotting the actual convergence delays we have shown the relative improvement (reduction) in the convergence delay in comparison to the case when we use the lowest of the three MRAI values (0.25 s). From the results we can see that for small failures, a low MRAI value results in the least convergence delay. However the situation is reversed for large failures and a higher MRAI value is more appropriate. Fig 6 shows the relative variation in the number of generated messages for the three different MRAI values. We again show the relative improvement (reduction) in the number of messages in comparison to the case when MRAI is 0.25 seconds. As expected, a higher MRAI always results in a lower number of messages. However the difference in the number of messages (for two different MRAI values) becomes more pronounced as the size of the failure is increased.

In Fig 7 we present the above results in a different way. Here we have plotted the convergence delay vs. the MRAI values for different failure magnitudes. If we look at any of the curves, we can see that as the MRAI is increased, the convergence delay first goes down and then increases. The increase in the convergence delay is mostly monotonic on both sides of the "optimal MRAI" (MRAI for which the convergence delay is the least). This is similar to the results observed by Griffin and Premore [8].

One of the factors responsible for the observed behavior and the "V" shaped curve is the processing overhead for BGP updates [8]. When MRAI is set equal to the optimal value, most if not all routers are able to process all received update messages during the MRAI period. Increasing the MRAI beyond the optimal MRAI means that the routers have to wait longer before sending the update messages and this increases the convergence delay. If we decrease the MRAI value, updates are generated at a faster rate and the processing load at the routers increases. So a router could possibly send out an update to a neighbor before it has processed all the queued update messages. If one of the remaining update messages changes the route which was just advertised, then another update needs to be sent. Not only does the neighbor have to process an extra update message, it might also send an extra update message

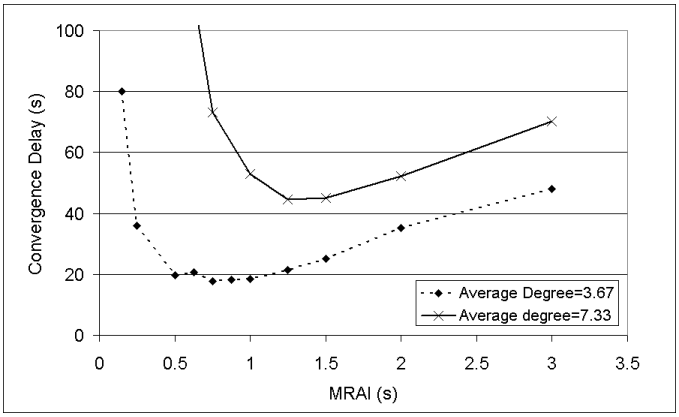


Fig. 9. Effect of average degree on convergence delay

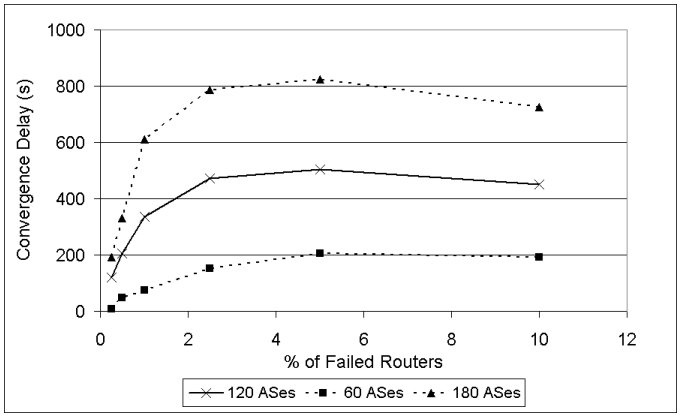


Fig. 10. Recovery time vs. Size of network (in terms of number of ASes)

to its peers, thus increasing workload on other downstream routers. This ultimately leads to higher convergence delay.

We have seen that larger failures result in more update messages which in turn lead to higher processing overhead. Thus for larger failures a larger MRAI value would be suitable, so that the routers have more time to process the extra messages . This can be seen in Fig 7 where an MRAI value of 0.25 seconds is ideal for 0.5% failure but less than the optimal value for 5% failures. For 5% failures the optimal MRAI is close to 1.5 seconds. Thus, not only is there no "optimal MRAI" value that works for all networks [8], *it is also not possible to choose an optimal MRAI value for a particular network (or even an AS) if we take failures of different magnitudes into account.* This result points to potential MRAI adjustment schemes based on the extent of failure. For example, one could set the MRAI to a low value (consistent with the expectation that most failures are small), and increase it in case of a large failure. This point is discussed in more detail later.

In Fig 8, we plot the the variation in the convergence delay for 2.5% failure vs. MRAI value for the three topologies: 70-30, 50-50 and 85-15 (all with an average degree of 3.67). As before, the first number (in an x-y distribution) refers to the percentage of low degree (degree 1-3) ASes. We can observe a distinct trend, and it is related to the degree of the high degree ASes in each of the topologies. In this particular scenario for example, the average inter-AS degrees of the high degree ASes in the 50-50 , 70-30 and 85-15 topologies are 5.3 , 7.6 and 13.1 respectively and the corresponding optimal MRAIs are roughly equal to 0.5, 0.75 and 1.0 seconds respectively. ASes with high degree are likely to receive the largest number of messages and hence the routers in those ASes are most likely to get overloaded. The higher the degree, the greater the processing load. Thus, for MRAI equal to 0.5 second, few routers in the 50-50 topology (high degree 5 or 6) seem to be overloaded, and the convergence delay is close to the minimum. But with the same MRAI, a larger number of routers in the 85-15 topology (high degree 13 or 14) can be expected to be overloaded, leading to a convergence delay significantly greater than the minimum value. We have to increase the MRAI to 1.0 second to remove the overload.

After looking at the effect of the degree distribution on the convergence delay vs. MRAI curve, we investigate the effect of the average degree on the same. In Fig 9, we plot the convergence delay (for 2.5% failure) for two topologies with the same type of degree distribution (70-30), but different average degree. One of the topologies is the same as the one that we saw in Fig 8, with average degree 3.67. In the other topology however, the high degree ASes have a degree of 19 or 20 resulting in an average degree of 7.33. We see that both the optimal MRAI and the convergence delay are greater for the topology with the higher degree. The larger optimal MRAI can be attributed to the greater degree of the high degree ASes, as we explained in the previous paragraph. The increase in the convergence delay is because of the greater number and length of alternate paths that have to be considered.

C. Network Size

In Fig 10 we show the effect of the size of the network on the convergence delay. We used the 70-30 degree distribution for these cases. As expected, we see that the convergence delay increases with the number of ASes in the network. That is because the number and the length of the routes go up with the size. The interesting thing to note here is that even if we keep the number of failed routers about the same, the convergence delay for a larger network is much higher. For example, a 1% failure in a 60 AS network incurs a convergence delay of about 75 seconds, but the convergence delay for a 0.5% failure in a 120 AS network is more than 200 seconds. Thus for large networks, even moderate sized area failures could result in long recovery times. Given the continued growth of the Internet, we expect that BGP recovery times will continue to increase. This clearly points to the need for stop-gap mechanisms that can avoid substantial packet losses or route resolution errors during the recovery process.

D. Distance-based Connections

As stated earlier, we did not consider the distance for deciding which ASes are directly connected by a link. In reality, routers connect preferentially to other routers that are nearby [22]. For small ASes, a similar property should

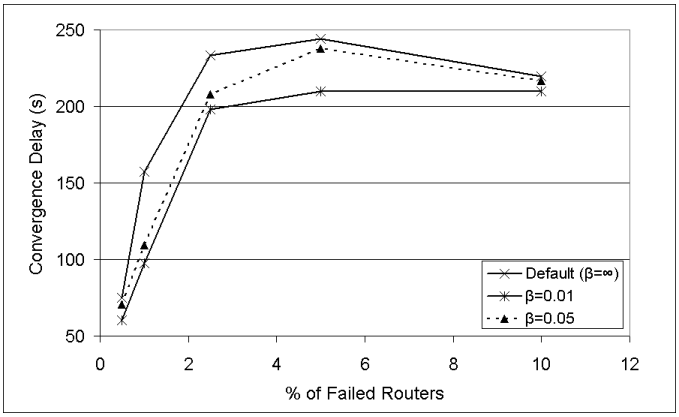


Fig. 11. Recovery time vs. distance based connectivity

hold with respect to AS-AS connectivity. For large ASes, the concept of a "nearby AS" may not be very meaningful because these ASes are spread over a large geographical area. However, for uniformity, we conducted experiments with distance based inter-AS connectivity where the inter-AS distance was defined to be the distance between the "center"s of the respective ASes. As the largest ASes cover almost the entire area of the map, their "location" will always be close to the center of the map. However, the heavy tailed distribution, which is used to generate the number of routers for each AS, ensures that the number of large ASes is small and hence the location is much more meaningful for the rest of the ASes. We used the Waxman connectivity scheme for creating the inter-AS edges. The probability that two ASes are connected was proportional to $e^{-d/\beta M}$ where d is the distance between the "locations" of the two ASes, M is the maximum possible distance and β is a dimensionless parameter. For our experiments we varied the values of β and observed the variation in the convergence delay.

For all the cases we used the same realistic degree distribution described earlier. The average inter-AS degree for the topologies was about 3.67. Fig 11 shows the results. It is clear that the convergence delay goes down as the decay rate β is decreased, i.e. as the probability of connecting to closer ASes is increased. The reason for the behavior is simple. A decrease in β leads to more links between geographically proximate ASes, and this means that these ASes now have less links connecting them to the rest of the network. The failure of a bunch of ASes in a contiguous area has less effect on the rest of the network, and hence the convergence delays go down.

E. Other Observations

In all the results that we have discussed till now, we considered a contiguous area of failure, as such failures are more likely. We did carry out a simulation run in which the failed routers were randomly distributed over the map. The maximum convergence delays for the distributed failure were found to be greater than that for the contiguous failure case. That is because in a contiguous failure, a number of the failed edges are between the failed routers (intra-AS edges are distance dependent) and hence do not have any effect on the convergence process. That is not the case with a distributed failure and hence the the overall effect is greater.

F. Summary

We now summarize the characteristics of the BGP convergence delay after large scale failures:

- The convergence delay initially increases and then goes down as the size of the failure is increased.
- Higher average inter-AS degree leads to higher convergence delays as well as a greater optimal MRAI.
- A small number of well connected ASes in a topology reduces the convergence delay (as compared to a case where all ASes have constant degree)
- The convergence delay first decreases and then increases as the MRAI is increased (First observed by Griffin and Premore [8]). The optimal MRAI for a failure increases with the size of the failure.
- The optimal MRAI for a particular topology depends on the degree of the best connected ASes.
- The convergence delay increases with the number of ASes in the topology.
- If ASes preferentially connect to nearby ASes, the convergence delay is reduced.

V. Reducing the Convergence Delay

In this section we present and analyze schemes designed to reduce the convergence delay. All these schemes are related to the MRAI parameter and are motivated by the fact that smaller MRAIs work best for small failures while larger MRAIs are more appropriate for large failures.

A. Degree Dependent MRAI

We have seen in the previous sections that the convergence process is closely linked to the behavior of the ASes with the highest degree. We have also seen that a high MRAI is more appropriate for larger failures. This leads to the idea of using a higher (than the rest of the ASes) MRAI at higher degree ASes. We still use a low MRAI at the ASes with low inter-AS degree so as to keep convergence delays low for small failures. Once again, 120 AS topologies with the realistic degree distribution were used for these experiments. In this distribution, a majority of the ASes have an Inter-AS degree in the range 1 to 3, and we use a low MRAI (0.25 seconds) at routers in these ASes. We used a high MRAI (2.0 seconds) at the rest of the ASes. This case is marked as (Low 0.25 s, High 2.0 s) in Fig 12. For comparison, we examined the reversed situation, i.e., MRAI=2.0 seconds in low degree ASes and MRAI=0.25 seconds in high degree ASes. This situation is marked as (Low 2.0 s, High 0.25 s) in Fig 12. Another case, in which all routers use the same MRAI of 2.0 seconds is also shown for comparison. In the figure we have again plotted the improvement in the convergence delay over that for MRAI=0.25 seconds.

From Fig 12 we can see that with the "Low 0.25 s, High 2.0 s" scheme, the convergence delay is decreased significantly for large scale failures (in comparison to the delay when the

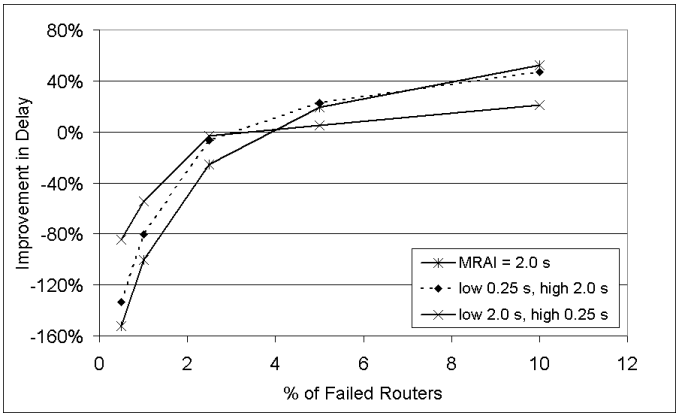


Fig. 12. Effect of degree dependent MRAI

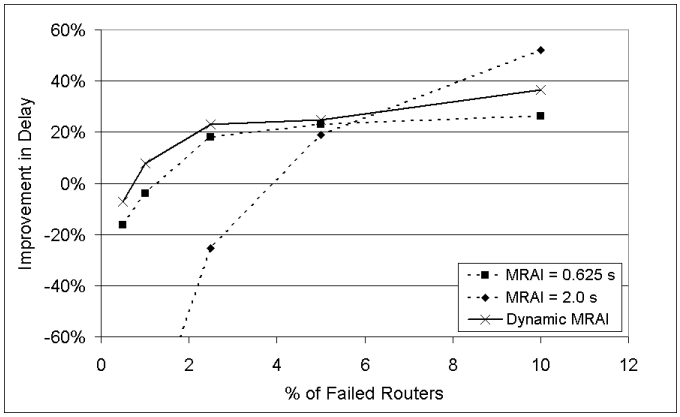


Fig. 13. Effect of Dynamic MRAI

MRAI=0.25 seconds at all routers). For small failures the "Low 0.25 s, High 2.0 s" scheme performs better than the case with constant MRAI of 2.0 seconds, but is still much worse than what we get with a constant MRAI of 0.25 seconds. The "Low 2.0 s, High 0.25 s" case, is not a good compromise either with high convergence delays for both small and large failures. Overall we observe that though the degree-dependent schemes offer some improvements over the constant MRAI cases, the convergence behavior is still strongly influenced by the MRAI used at the high degree ASes. This deficiency can be addressed by changing the MRAI dynamically, as discussed next.

B. Dynamic MRAI

Going back to what we mentioned in the beginning of this section, we would like to use low MRAIs for small failures, and higher MRAIs for larger failures. As small failures are more common, we can set the default MRAI to a low value, and switch to a higher MRAI if needed. Thus, if we have a scheme that can quickly determine the size of the failure and set the MRAI accordingly, we can minimize the convergence delay for failures of different magnitudes. However such a scheme would probably require data to be collected from multiple ASes and hence might add significant overhead to the BGP convergence process. We therefore decided to focus on schemes that can operate independently inside an AS.

Our Dynamic MRAI scheme focuses on the effects of a failure. Just like the convergence delay, the number of messages also increase with the size of a failure. As mentioned earlier, if the update messages cannot be processed during the MRAI period, then this will cause the generation of extra updates which in turn will lead to even more multiple overloaded routers and large convergence delays. We therefore use the number of queued update messages at a router as the criterion to determine if we should modify the MRAI at that router. If a router is overloaded, increasing the MRAI at that router will not only reduce the number of update messages it generates but will also cut down the number of invalid routes that it sends to its neighbors. So, this type of scheme can reduce the convergence delay by reducing the overall processing overhead in the network and by decreasing the number of invalid routes during the convergence process.

We implemented a scheme in which we varied the MRAI at a router between three different values. From the observed convergence delays for 120 AS networks with realistic degree distributions we chose the values 0.25, 0.625 and 2.0 seconds. The selection was based on the observations that MRAI equal to 0.25 seconds resulted in the least convergence delay for small (0.5-1%) failures, while MRAI equal to 0.625 seconds was ideal for 2.5% failures and 2.0 seconds was good for failures in the 5 to 10% range. The MRAI is set to the lowest value (0.25 seconds) in the beginning because small failures are much more likely and in that scenario we will automatically incur the least delay. In our scheme, we monitor the queue length of update messages as an indicator of overload. We convert the queue length into *unfinished work* by multiplying it by the average processing delay. If the unfinished work is greater than a threshold (*upTh*), then we increase the MRAI if possible. If the unfinished work is less than another threshold (*downTh*), then we decrease the MRAI if possible. It must be noted that even if we decide to change the MRAI, we do not modify the values of the running timers; instead, the change takes effect only when the timers are restarted after an update has been sent. We did this to keep the implementation simple. In our experiments we configured the average processing delay. In a real system, the average processing delay can either be computed at the router itself, or configured by the operator if that is deemed more appropriate.

We show the effects of using this Dynamic MRAI scheme in Fig 13. We have again plotted the improvement in the convergence delay over that for MRAI=0.25 seconds in Fig 14. For this set of results we set the *downTh* to 0.05 seconds and the *upTh* to 0.65 seconds. We can see that the Dynamic MRAI scheme performs quite well. The convergence delay for small (0.5-1%) failures is close to that with MRAI=0.25 seconds. For 2.5% failure, the convergence delay for the dynamic scheme is better than that for MRAI=0.625 seconds. For larger failures, the delays for the dynamic scheme are a bit worse than the convergence delay for MRAI=2.0 seconds but significantly better than that for MRAI=0.25 seconds. Thus we see that with this dynamic scheme, we were able to achieve close to minimum convergence delay for a wide range of failures.

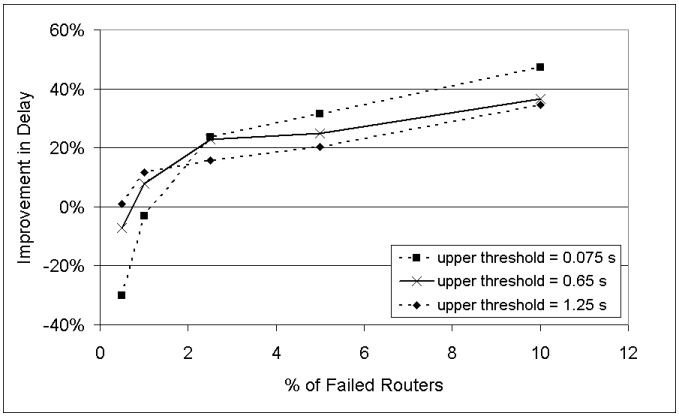


Fig. 14. Effect of *upTh* on convergence delay

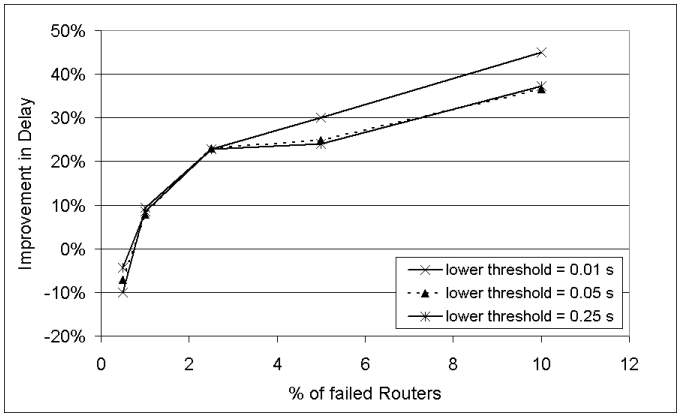


Fig. 15. Effect of *downTh* on convergence delay

We also found that the number of messages generated by the Dynamic MRAI scheme marginally greater than what we get if we use an MRAI of 2.0 seconds. We tested this scheme for topologies with 240 ASes as well. We obviously had to change the MRAI values but we kept the thresholds the same. The results were again very good and similar to what we have shown here, and are omitted to avoid repetition.

Now we look at the performance of the dynamic scheme if the thresholds are varied. We first set *downTh* to 0.05 and experimented with a number of *upTh* values. If *upTh* is low, then the behavior is similar to having a constant high MRAI, because too many routers increase their MRAI. Thus we see that with a low threshold, the convergence delay for small failures is comparatively high, but the delays for large failures are low. As we increase the threshold, fewer routers increase their MRAIs. Hence the convergence delays for small failures go down but the delays for the larger failures go up. However we see that increasing the *upTh* to 1.25 seconds from 0.65 seconds doesn't have a drastic impact on the convergence delay, and we are able to get good results for a range of *upTh* values.

Next we have a look at the effect of the *downTh* value on the delays. We show the results for those experiments in Fig 15. Here we have set *upTh* to 0.65 seconds. As we decrease *downTh*, routers use a high MRAI value for a longer duration and we observe comparatively higher convergence delays for smaller failures and comparatively lower delays for larger failures. We again observe similar results for a range of values (0.05-0.25). Thus we can see that, although the thresholds are an integral component of the dynamic scheme, the performance is not very sensitive to the values for the thresholds. Hence, we do not need to worry about selecting the absolute optimum values for the thresholds.

We reran the experiments with the dynamic scheme implemented at the routers in high degree ASes only to see how the results are affected. We have seen in the previous section that the convergence delay for large failures is heavily dependent on the MRAI of the routers in high degree ASes, and therefore it made sense to change the MRAI only in those ASes. However we found that the results were effectively the same as when we had the dynamic scheme at all the ASes. This was because the routers in low degree ASes rarely (if ever) got overloaded and hence the MRAI at those routers stayed at the minimum value. Thus we can get significant improvements in the convergence delay, even with a partial deployment of this scheme by a few large ISPs. We also tested out some other schemes for dynamically varying the MRAI. In the first scheme, we used the processor utilization to detect overload and to change the MRAI. We got promising results with that scheme as well. In the second scheme, we monitored the number of update messages received at a router. This scheme was not very successful as it was difficult to set the up and down thresholds.

The Dynamic MRAI scheme does have a couple of deficiencies. The first one has to do with the selection of the MRAI values, as these are dependent on the network. For our experiments we measured the convergence delays for different MRAI values, and then picked the MRAIs that resulted in the least delay for different failure magnitudes. This empirical approach is viable for small or moderate sized networks, but for large networks like the Internet (more then 20,000 ASes) one will have to estimate the MRAI values. We are currently looking at the theoretical basis for the selection of the parameters. Secondly, with the dynamic scheme, the convergence delay for large failures is somewhat higher than that with the largest MRAI (2.0 seconds in this case). There are a couple of reasons for this. First, all routers start off with the lowest MRAI (0.25 seconds) and it takes a while for the queues at the overloaded routers to exceed the *upTh*. Second, the MRAI change takes effect only after the timer expires. We are exploring ways to reduce the response time and improve this aspect of the scheme.

C. Batching of Update Processing

The default implementation of BGP processes all messages in the FIFO order and this may result in the generation of invalid updates and unnecessary processing of some messages. As an example, suppose that router *A* sends an update to neighbor *B* at time *t* and at that time there are four pending update messages in the queue. The first and third messages advertise a new route for destination *X* while the second and fourth messages advertise a new route for destination *Y*. Let's

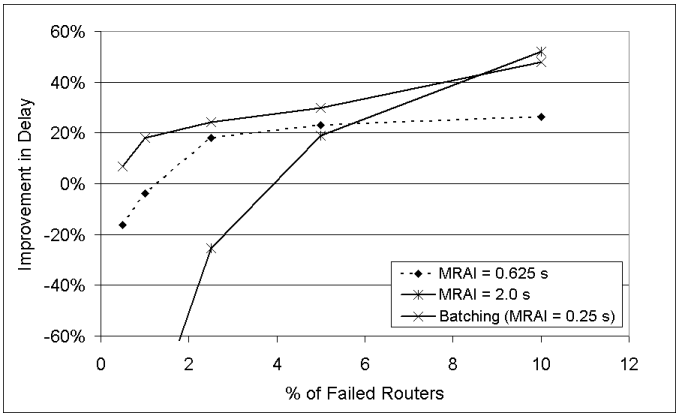


Fig. 16. Performance of Batching Scheme

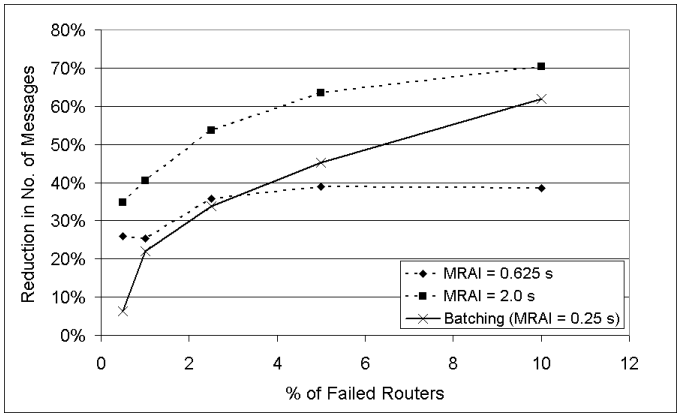


Fig. 17. Number of messages generated by the Batching scheme

also assume that each update results in a new best route for the corresponding destination, and that none of these routes pass through B. The updates will be processed in FIFO order by default. If the MRAI timer expires before the last two messages have been processed, then *A* will send two updates to *B* (one each for *X* and *Y*). Two more updates will be sent out after the final two updates have been processed. So, in all four updates will be sent from *A* to *B*. However, if the timer expires after all the update messages had been processed, then only two update messages will be generated. We can rectify this situation by simply reordering the messages in the queue. For example, if we move the third message (for destination *X*) to the second position, then both the update messages for *X* will be processed before the MRAI timer expires. So one update message (for *X*) will be sent after the timer expires, and another one (for *Y*) will be sent after the final two messages are processed. This leads to the idea of batched processing.

In the batched processing scheme, we effectively maintain a separate logical queue/batch for each destination. When an update arrives we extract the destination, and queue it appropriately. Even with a large number of destinations, this can be implemented efficiently using hashing. The updates are processed according to the time of arrival of the first message in the batch. As a result we can process all updates for a destination together and thereby address the problem identified above. Furthermore, we can delete multiple update messages from the same neighbor, as the older updates are now invalid. The price of destination based queuing should be small as compared to the benefits achieved.

We show the performance for the Batching scheme in Figs 16 and 17. For the Batching scheme, we set the MRAI to 0.25 seconds. We observe that the Batching scheme is able to reduce the convergence delay for larger failures significantly while keeping the delays low for small failures. If we combine the Batching and Dynamic MRAI schemes, then we are able to decrease the delays even further. The primary aim of the Batching scheme is to reduce the number of updates generated by overloaded routers. As shown in Fig 17, the number of messages for larger failures is much less than that with MRAI=0.25 seconds and is in the same range as the number of messages for MRAI=2.0 seconds.

We also carried out experiments to observe the effect of the Batching scheme with other MRAI values. We show the convergence delay for 5% failure, with different MRAIs in Fig 18. We observe that the convergence delay decreases significantly with Batching if the MRAI is less than the optimal value; however Batching does not have much of an impact otherwise. This is to be expected because the Batching scheme is effective only when there are overloaded routers in the network. If the queue of update messages is small, batching might not be possible at all. Even if some messages are rearranged, the effect is unlikely to be significant.

On a separate note, another form of "batching" is carried out in BGP routers today. This is done to mitigate the speed mismatch between the rate at which BGP updates can be processed (fast) and the rate at which the new routes can be transferred to the line cards (slow). Typically one data buffer (TCP) is read from each peer connection and all the collected BGP updates are processed sequentially in a batch, after which the route changes are transmitted to the line cards. During periods of overload this scheme can provide some of the same benefits as our scheme, if two updates for the same destination are present in the same batch. If the size of the failure is large however, the number of destinations for which updates are sent will be high, and the probability of having two updates for the same destination in a batch will progressively decrease. Thus our scheme should perform much better for large scale failures.

D. Comparison with Other Schemes

In this section we compare the performance of our schemes with that of two well known BGP variants, Ghost Flushing [13] and Consistency Assertions [14]. For this purpose we use 120 node topologies with just one router per AS, as similar topologies were used by the authors of Ghost Flushing (GF) and Consistency Assertions (CA) to demonstrate the efficiency of those schemes. We used the realistic degree distribution to generate the inter-AS links. We first collected the convergence delays for 1-10% failures in this type of topology, using multiple MRAIs. After determining that an MRAI of 0.5 seconds was ideal for the smallest failures (1%), we used that as the MRAI for all the schemes. For the Dynamic MRAI

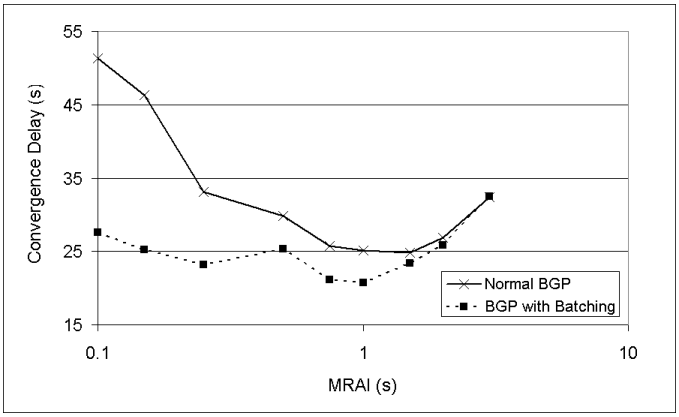


Fig. 18. Effect of Batching with different MRAIs

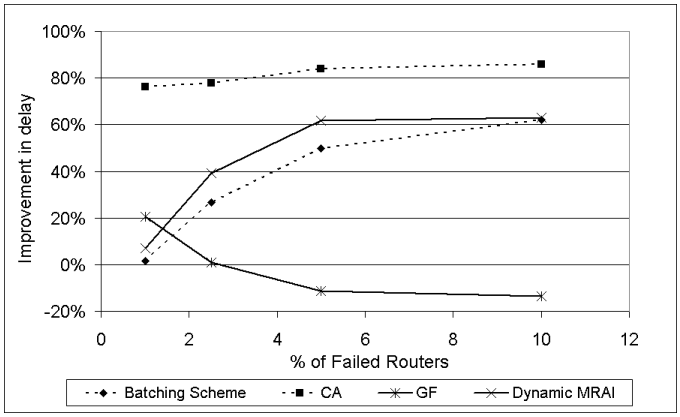


Fig. 19. Comparison with other BGP Variants

scheme, we chose MRAIs of 0.5, 1.0 and 3.0 seconds from the observed results. The improvement in the convergence delays, as compared to normal BGP with MRAI = 0.5 seconds, is shown in Fig 19. We see that Ghost Flushing does not do very well, especially for larger failures. Ghost Flushing increases the number of messages that are generated, and this makes things worse when routers are overloaded after large failures. We see that Consistency Assertions has the best performance, while Dynamic MRAI and Batching also provide significant improvements in the convergence delay. Unlike Consistency Assertions, our schemes do not modify the best path selection algorithm of BGP and hence the convergence delay for a failure of a particular magnitude is bounded at the lower end by the optimal convergence delay for normal BGP at that magnitude. It must be noted however that Consistency Assertions makes the assumption that a router cannot advertise two different routes to a particular destination, and that is no longer true in the current Internet. Outbound policies that modify the advertised path and send different paths to different neighbors are quite common.

As our schemes do not modify the best path selection or the update generation algorithm for BGP, they can be combined with many other BGP variants including Ghost Flushing and Concistency Assertions. In Fig 20, we show the additional improvement in the convergence delay when we combine the Batching scheme with Ghost Flushing and Consistency Assertions. We are able to get major improvement over Ghost Flushing as the original (for Ghost Flushing) convergence delays were worse than even normal BGP. The convergence delays for Consistency Assertions were already much better than normal BGP, but we were able to reduce those by up to 20% when we combined the Batching scheme. We can get similar results by combining Ghost Flushing and Consistency Assertions with an appropriately configured Dynamic MRAI scheme.

In conclusion, we have established that the Batching scheme as well as the Dynamic MRAI scheme can minimize the impact of large scale failures substantially by reducing the convergence delay, without increasing the recovery times for small failures. Furthermore these schemes can be combined with other BGP variants to decrease the convergence delays even further.

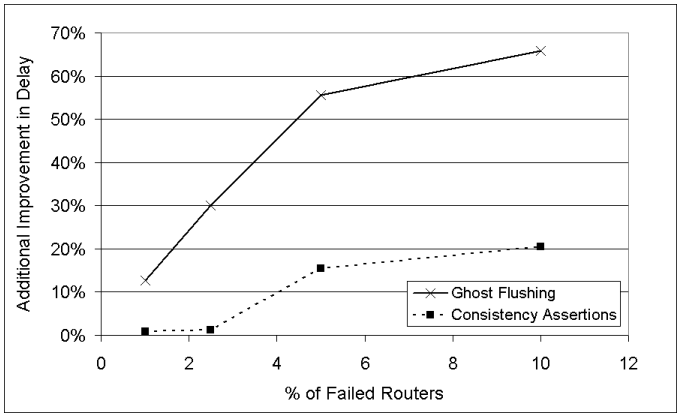


Fig. 20. Improvement in delay after combining Batching with other schemes

VI. Conclusions

This study sheds light on how inter-domain routing in the Internet will behave under large-scale failures. We found that, initially the BGP convergence delay increases rapidly with the magnitude of failure before levelling off and going down. This means that multiple failures can lead to much longer periods of instability as compared to single failures. Furthermore, even with a fixed number of failed routers, the recovery time increases as the size of the network goes up. Therefore, the convergence delay for large scale failures in the Internet can be expected to keep increasing in the future. The paper also points to other important aspects about BGP convergence delay. In particular, a heavy tailed distribution for inter-AS connectivity (which is present in the Internet today) and distance based connectivity (which is highly likely to exist in the Internet) help in bringing down the delay. Also, the degree distribution seems to have a stronger influence on the convergence delay than distance based connectivity.

We took a detailed look at the effect of BGP's MRAI (Minimum Route Advertisement Interval) parameter on the convergence delay for large scale failures. We found that the MRAI has a significant effect on the shape of the convergence delay vs. magnitude of failure curve. We discovered that the

"optimal" MRAI, or the MRAI value at which we incur the least convergence delay, is dependent on the size of the failure and actually increases with the size. Thus, there is no single MRAI value which will provide the best convergence delay for different types of failures in a network. We also observed that the "optimal" MRAI is dependent on the degree distribution of the network. We investigated the effect of having different MRAIs at different routers and we saw that the convergence delay for larger failures is dependent on the MRAI of the routers in higher degree ASes.

We presented a dynamic scheme to vary the MRAI at a BGP router. This scheme automatically tries to select the "optimal" MRAI for a failure, based on the size of the message queue at the router. We found that the dynamic scheme worked very well, and the convergence delay was always close to the minimum for failures of various magnitudes. The dynamic scheme reduced the convergence delays for large scale failures while keeping the delays low for smaller, more probable failures. The parameters for this scheme were the three different MRAI values and the two thresholds, all selected based on experimental results. In order to use this type of scheme in real networks, it is necessary to develop a suitable theory for choosing various parameters. This work is currently ongoing. We also examined a Batching scheme, designed to reduce the generation of invalid route advertisements and to remove stale update messages, during periods of overload. We found that the Batching scheme can substantially cut down the convergence delays. Another advantage of the Batching scheme is that it does not use any configuration parameters. Finally we combined the Batching scheme with two other BGP variants, Ghost Flushing and Consistency Assertions, and were able to improve the convergence delays even further.

Both of our proposed schemes are designed to improve the convergence delay in situations where the update processing load at BGP routers is high. If the processing delays are so small that the BGP routers do not get overloaded, then the convergence delays will be unchanged. However the processing load is also dependent on the number of update messages which in turn depends on the number of destinations affected by the failure. Despite the advances in router processor speeds, a large scale failure in the Internet, which contains nearly 200,000 destinations, will generate a huge number of updates that is likely to overwhelm a large number of routers. Hence our schemes will be effective in such a scenario. At the same time, these schemes provide the opportunity for service providers/network operators to safely decrease the default MRAI, so that the convergence delays for small failures can be reduced, while keeping the convergence delays for large failures in check. The default MRAI in the Internet is set to 30 seconds primarily to guard against a network meltdown in case of a large failure, and our schemes can enable this default value to be lowered.

REFERENCES

[1] A. Sahoo, K. Kant, and P. Mohapatra, "Characterization of BGP recovery under Large-scale Failures," in *Proc. ICC 2006*, Istanbul, Turkey, June 11–15, 2006.

[2] A. Sahoo, K. Kant, and P. Mohapatra, "Improving BGP Convergence Delay for Large Scale Failures," in Proc. DSN 2006, June 2528, 2006, Philadelphia, Pennsylvania, pp. 323332.

[3] Y. Rekhter, T. Li, and S. Hares, "Border Gateway Protocol 4," RFC 4271, Jan. 2006.

[4] Bassam Halabi, *Internet Routing Architectures*, 2nd Ed. Cisco Press, 2000.

[5] C. Labovitz, G. R. Malan, and F. Jahanian, "Internet Routing Instability," *IEEE/ACM Transactions on Networking*, vol. 6, no. 5, pp. 515–528, Oct. 1998.

[6] Labovitz, C., Ahuja, et al., "Delayed internet routing convergence," in *Proc. ACM SIGCOMM 2000*, Stockholm, Sweden, Aug. 28–Sep. 1, 2000, pp. 175–187.

[7] Dan Pei, B. Zhang, et al., "An analysis of convergence delay in path vector routing protocols," *Computer Networks*, vol. 30, no. 3, Feb. 2006, pp. 398–421.

[8] T.G. Griffin and B.J. Premore, "An experimental analysis of BGP convergence time," in *Proc. ICNP 2001*, Riverside, California, Nov. 11–14, 2001, pp. 53–61.

[9] D. Obradovic, "Real-time Model and Convergence Time of BGP," in *Proc. IEEE INFOCOM 2002*, vol. 2, New York, Jun. 23–27, 2002, pp. 893–901.

[10] R. Teixeira, S. Agarwal, and J. Rexford, "BGP routing changes: Merging views from two ISPs," *ACM SIGCOMM Computer Communications Review*, vol. 35, issue 5, pp. 79–82, Oct. 2005.

[11] J. Rexford, J. Wang, et al., "BGP routing stability of popular destinations," in *Proc. Internet Measurement Workshop 2002*, Marseille, France, Nov. 6–8, 2002, pp. 197–202.

[12] C. Labovitz, A. Ahuja, et al., "The Impact of Internet Policy and Topology on Delayed Routing Convergence," in *Proc. IEEE INFOCOM 2001*, vol. 1, Anchorage, Alaska, Apr. 22—26, 2001, pp. 537–546.

[13] A. Bremler-Barr, Y. Afek, and S. Schwarz, "Improved BGP convergence via ghost flushing," in *Proc. IEEE INFOCOM 2003*, vol. 2, San Francisco, CA, Mar. 30–Apr. 3, 2003, pp. 927–937.

[14] D. Pei, X. Zhao, et al., "Improving BGP convergence through consistency assertions," in *Proc. IEEE INFOCOM 2002*, vol. 2, New York, NY, June 23–27, 2002, pp. 902–911.

[15] S. Deshpande and B. Sikdar,"On the Impact of Route Processing and MRAI Timers on BGP Convergence Times," in *Proc. GLOBECOM 2004*, Vol. 2, pp 1147- 1151.

[16] W. Sun, Z. M. Mao, K. G. Shin, "Differentiated BGP Update Processing for Improved Routing Convergence," in *Proc. ICNP 2006*, Santa Barbara, California, November 12–15, 2006 , pp. 280–289.

[17] A. Medina, A. Lakhina, et al., "Brite: Universal topology generation from a user's perspective," in *Proc. MASCOTS 2001*, Cincinnati, Ohio, August 15–18, 2001, pp. 346-353.

[18] "SSFNet: Scalable Simulation Framework". [Online]. Available: http://www.ssfnet.org/

[19] B. Waxman, "Routing of Multipoint Connections," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, pp. 1617–1622, Dec. 1988.

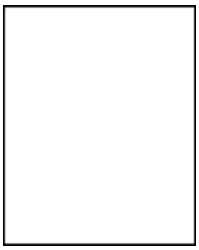
[20] A.L. Barabasi and R. Albert, "Emergence of Scaling in Random Networks," *Science*, pp. 509–512, Oct. 1999.

[21] T. Bu and D. Towsley, "On Distinguishing between Internet Power Law Topology Generators," in *Proc. IEEE INFOCOM 2002*, vol. 2, New York, Jun. 23–27, 2002, pp. 638–647.

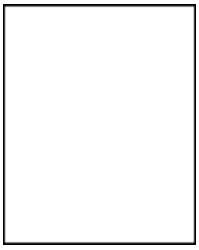
[22] A. Lakhina, J.W. Byers, et al., "On the Geographic Location of Internet Resources," *IEEE Journal on Selected Areas in Communications*, vol. 21 , no. 6, pp. 934–948, Aug. 2003.

[23] H. Tangmunarunkit, J. Doyle, et al, "Does Size Determine Degree in AS Topology?," *ACM SIGCOMM Computer Communication Review*, vol. 31, issue 5, pp. 7–10, Oct. 2001.

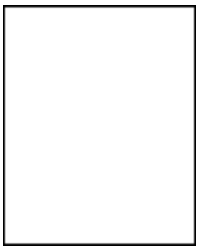
[24] B. Zhang, R. Liu, et al., "Measuring the internet's vital statistics: Collecting the internet AS-level topology ," *ACM SIGCOMM Computer Communication Review*, vol. 35, issue 1, pp. 53–61, Jan. 2005.
Latin American networks," in *Proc. LANC 2003*, La Paz, Bolivia, Oct. 4–5, 2003, pp. 35–43.



Amit Sahoo received his Masters degree in Computer Science from Michigan State University in 2002. He is currently working towards a Ph.D. in Computer Science at the University of California, Davis. His current research deals with the analysis and improvement of the reconvergence process of the Border Gateway Protocol. He has previously worked on overlay networks, simulation of distributed clusters, and QoS mechanisms at the transport layer.



Krishna Kant has been with Intel corporation since 1997. His current research interests are in resource management issues in virtualized data centers, robustness of Internet infrastructure, and fabric features required for advanced data centers. From 1992-1997 he was with Bellcore working SS7 congestion control. From 1981 to 1992 he held academic positions in Northwestern University and Pennsylvania State University.



Prasant Mohapatra is currently a Professor in the Department of Computer Science at the University of California, Davis. In the past, he was on the faculty at Iowa State University and Michigan State University. Dr. Mohapatra received his Ph.D. in Computer Engineering from the Pennsylvania State University in 1993. He was/is on the editorial board of the IEEE Transactions on computers, IEEE Transaction on Parallel and Distributed Systems, ACM WINET, and Ad Hoc Networks. He has been on the program/organizational committees of several international conferences. Dr. Mohapatra's research interests are in the areas of wireless networks, sensor networks, Internet protocols and QoS.