BGP Rerouting Solutions for Transient Routing Failures and Loops

Jian Qiu, Feng Wang and Lixin Gao
Dept. of ECE, University of Massachusetts, Amherst, MA 01003
{jqiu, fewang, lgao}@ecs.umass.edu

***Abstract*— During the routing convergence processes of BGP system, the end-to-end reachability can be temporarily disrupted due to transient routing failures or loops in the forwarding paths. This could lead to severe performance degradation and even service disruption, especially for the real-time interactive applications. In this paper, we explore feasible modifications of BGP to eliminate transient routing failures and loops. First, we find that the existing BGP convergence acceleration solutions, such as ghost-flushing and EPIC, can eliminate transient forwarding loops but exacerbate transient routing failures. Then we propose an indicative re-routing scheme, which enable BGP to piggyback an indicator of alternative paths with each route, to improve route visibility and thus eliminate transient routing failures. However, we find that it might worsen transient forwarding loops at the same time. Finally, we exploit the synergy of the combination of the two types of schemes and propose an indicative+EPIC re-routing scheme. It is found capable of eliminating both transient forwarding failures and loops.**

I. Introduction

The Internet has been evolving into a worldwide information infrastructure. Newly emerging real-time IP-based services, such as Web TVs and interactive games, are progressively deployed while mission-critical applications, such as voice services and virtual private networks, are increasingly moving from the traditional telecommunication network to the Internet. However, compare with the traditional network which offers over 99.99% availability, the reliability of the Internet is still far from satisfaction. Packet loss and forwarding loops are common [10], [14]. Measurement work shows that transient failure in the Inter-domain routing system is a major reason leading to Internet-wide end-to-end performance degradation [7], [15], [21].

The Border Gateway Protocol (BGP) [18] is the only Internet routing protocol deployed in the Internet. Labovitz *et al* [8] find that the BGP system can take as long as 30 minutes to converge to a stable state once a destination network disconnects from the Internet. Even worse, during the convergence process triggered by a failover event, in which the network failures do not disconnect destination from the network but force nodes to choose less preferable paths, packet forwarding loops are observed [7], [15]. Further, Wang *et al* [21] found that some nodes can temporarily lose routes to the destination despite the destination is physically reachable, as is referred to *transient routing failures*. As a result, during the failover convergence process, packets sent to destination can be dropped due to either transient forwarding loops or routing failures. The loops or failures can last as long as tens of seconds, which can drastically degrade end-to-end performance and even cause service disruptions.

In this paper, we take the endeavor to explore feasible BGP modification to eliminate the transient loops and failures during the BGP failover convergence process. Our major findings are summarized as follows.

At first, the existing BGP convergence acceleration solution, such as ghost-flushing [4] and EPIC [5], are able to eliminate transient forwarding loops in failover convergence process. However, they exaggerate transient routing failures.

Second, inspired by the multi-path re-routing scheme [2], which enable BGP exchange more than one path between routers, we propose an indicative re-routing scheme, which piggyback indicators of alternative routes with ordinary routes to inform the existence of alternative routes. During failover convergence process, when a node's valid routes are temporarily removed, it can use these indicators as hints of possible alternative routes and forward packets to the relevant neighbors. The experiment shows the indicative re-routing scheme can completely eliminate transient routing failures. Nonetheless, it might exacerbate transient forwarding loops.

Finally, we leverage the advantages of the two types of schemes and propose an indicative+EPIC re-routing scheme, in which the indicative re-routing provides alternative routes to nodes to eliminate failures while the EPIC remove obsolete routes to avoid loops. We built a simulator to examine the performance of our solutions. The results confirm that the combined solution can eliminate both transient failures and loops.

The rest of the paper is organized as follows. Section 2 introduces the backgrounds. Section 3 elaborates the solutions for transient forwarding loops, for transient routing failures and for both respectively. In section 4, simulations results are presented. Finally, section 5 reviews the related work and section 6 concludes the paper.

II. Backgrounds

A. BGP Model

A BGP system is modeled as a graph $G = (V, E)$, where the node set V consists of all BGP routers and the edge set E is composed of all BGP peering sessions. Without loss of generality, node 0 is designated as the destination node. BGP routers are grouped into *Autonomous Systems (ASs)*. In order to ensure route consistency and avoid routing loops within an AS, the BGP routers in an AS must be fully connected, or in a 2-tiered structure, in which the backbone routers, acting as route reflector servers, are fully meshed while the edge routers, which peer with neighboring ASs, connect to the relevant backbone routers. The peering sessions between routers in the same AS are called *iBGP* and those between distinct ASs are *eBGP*.

As a path vector protocol, BGP's route is represented with a simple path consists of ASs. The AS path indicates the reverse AS sequence along which the route is propagated. In BGP system, a router u maintains two types of route information bases (RIB): rib_in and loc_rib. It learns routes from its neighbors and stores them in rib_in. Among these routes, it choose one best route for each destination and stores in loc_rib. For a destination p, once u detects the change of its best route, u advertises an *announcement* to tell its neighbors its choice. When u finds its best route becomes unavailable, it first attempts to find an alternative one in its rib_in. If none is found, u sends a *withdrawal* to its neighbors to indicates the absence of valid routes to p. In addition, current BGP implementations employ a mechanism called *poison reverse*, in which when a node u installs a route in its loc_rib, it sends a withdrawal instead of an announcement to its predecessor from whom u learns this route.

When forwarding packets destined to a destination, a node u uses its best route in its loc_rib to determine the next-hop neighbor to whom it should forward. At a time point t, the *forwarding path* of u to the destination is the sequence of routes through which the packets actually traverse from u to the destination. The forwarding path of u at t can be constructed by starting from u and recursively appending the next-hop to the destination at each router sequentially until the destination is reached.

B. Routing Delay

Suppose that a router u updates its loc_rib and sends a routing update to one of its neighbors, say v, the interval between u and v's relevant RIB update operation is defined as *routing delay*, which includes queuing delay, transmission delay, and route information processing delay. We use d to represent the upper bound of this one-hop delay.

In addition, in order to prevent update messages from overwhelming the network, the *Minimal Route Advertisement Interval* (MRAI) timer is employed to regulate the minimal rate between consecutive routing updates sent from a node to its relevant neighbors. We use D to represent the upper bound of the MRAI timer. If a message is delayed by the MRAI timer, its one hop routing delay is artificially prolonged and upper bounded by $D + d$. Since $D \gg d$, the timer-induced routing delay is approximately D.

There are various MRAI implementation variances. Typically, the MRAI timer is applied to regulate the announcements only. However, the MRAI timer can space out the advertisement of withdrawal messages either, as is referred to as *withdrawal rate-limiting* (WRATE) and standardized in the BGP specification [18]. Empirical study [15] shows that WRATE can reduce the number of routing update messages while prolonging the convergence process. Nonetheless, the reduction becomes almost negligible but the prolongation is significant when WRATE is applied in an Internet-like topology. Thus we suggest the MRAI timer be applied to announcements only. At the same time, although the MRAI timer delays the routing propagation, it is controversial to completely disable MRAI timer since it helps mitigate overhead of routing updates. Therefore we suggest the MRAI timer be employed.

C. BGP Convergence

In the BGP system, if no router changes its loc_rib, we say the system reaches its stable state. Once the topology changes, such as link/node failure or recovery, the relevant routers might adjust their routes by exchanging routing updates until the system converges to another stable state. The process is named as *convergence process*.

BGP convergence process can be triggered by various events. Basically, in terms of the topological changes, there are two types of events: *failure* or *recovery*. Failure makes some nodes/links unavailable and recovery brings back some nodes/links into the system. In addition, the impact of failures on the reachability of the destination can be: (1) *Faildown*: the failures disconnect the destination from the network, and (2) *Failover*: the failures force nodes to use alternative routes to reach the destination.

Measurements show that the BGP convergence processes triggered by failures can last as long as 30 minutes, as known as *BGP slow convergence* [8]. The basic causes of slow convergence is that during the convergence process each node literally has to install several ”ghost” routes that traverse the failed components before it finally capture the valid routes. In addition, the MRAI timer further lengthens the process by delaying the propagation of routing updates. It shows that the convergence process takes $O(D)$. Several solutions have been proposed. They share the idea that employs certain mechanisms to accelerate the removal of ”ghost” routes. For instance, in ghost-flushing scheme [4], whenever a node's current best path is replaced by a less preferred route, the node immediately sends withdrawal messages to all its neighbors to remove the ”ghost” routes containing the removed path. A more sophisticated approach is proposed in the EPIC scheme [5], in which, after a link failure, nodes relies on the *fesnList* piggybacked with every routes, which implicitly indicates the location of the failed link, to remove the ”ghost” routes.

It shows that both the ghost-flushing and the EPIC can accelerate the BGP convergence process triggered by faildown events from $O(D)$ to $O(d)$. However, for the convergence process triggered by failover events, in which infected nodes finally converge to alternative routes, they shows poor performance because both solutions do their best to accelerate the removal of ”ghost” routes while doing nothing to speed up the dissemination of the alternative routes. Note that the failover events is distinct from the faildown events in the sense that the destinations in the former scenario are physically reachable. Thus, the top priority of a BGP convergence acceleration solution for the failover events should accelerate not only

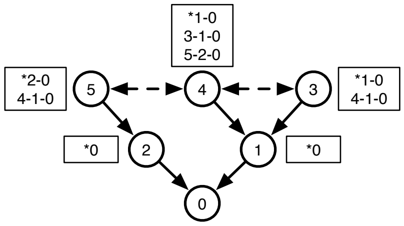


Fig. 1. Example of transient failures

the removal the ”ghost” routes but also the acquisition of alternative routes such that the reachability of the destination will not be disrupted. From this viewpoint, we should evaluate a convergence acceleration solution for the failover events in terms of end-to-end reachability continuity but not the convergence speed.

D. *Transient Failures and Loops*

After a failover event, if a node u loses routes to the destination, i.e. its loc_rib becomes empty, u is said to experience *transient routing failure*. At the same time, even if a node has a route, its reachability to the destination can be disrupted due to the failures or loops of its forwarding path. In the former case, in which its forwarding path stops at a node in transient routing failure, as named as *transient forwarding failure* and in the latter case, in which its forwarding path forms a loop, as called *transient forwarding loop*. Note that although both transient *failures* and *loops* disrupt the reachability to the destination, the disruptions are caused by distinct reasons: in the former packets are dropped because the relevant nodes have no route to the destination while in the latter the packet drop is because of running out of TTL.

We use an example to illustrate how the transient failures and loops are formed during the failover convergence process. As shown in Figure 1, we assume that each node represents an AS and every node chooses the shortest route in terms of AS path length as the best route and prefer the route from a lower id node if there is a tie. The routing table of each node is shown in the box besides the node, in which route entries are sorted in descending order of preference. Suppose that the link between 0 and 1 breaks. Once node 1 detects the failure and remove its best path 1-0 from loc_rib, its loc_rib becomes empty and it experiences transient routing failure. Then 1 sends withdrawal to 3 and 4. Before the arrival of the messages, 3 and 4 keep sending packets to 0 to 1. During the period, 4 and 4 experience transient forwarding failures. After 3 and 4 receive 1’ withdrawal, they will instantly adopt their next-preferred alternative routes, e.g. 3-4-1-0 and 4-3-1-0 respectively. After that, before they know each other’s new routes, the packets sent by either 3 or 4 to 0 will go through a forwarding loop 3-4-3-$\ldots$. The transient forwarding loops continues until 3 or 4 receives the other’s message.

III. Solutions for transient failures and loops

In order to eliminate reachability disruption caused by transient failures or loops during the failover convergence process, we need to address the failures and loops simultaneously. The transient failures, including transient routing failures and transient forwarding failures, result from the temporary absence of valid routes at nodes due to the constraints from both the protocol and the network topology. The transient loops are caused by the inconsistency of routes across nodes. Therefore, we need to employ different strategies to eliminate or reduce transient failures and loops separately. We first examine the solutions for transient forwarding loops and then investigate the solutions for transient failures. Because transient forwarding failures are basically caused by the transient routing failures of certain nodes on the forwarding path, we will only address the transient routing failures.

A. *Solutions for transient forwarding loops*

The basic reason of transient forwarding loops is routing inconsistency across nodes. For example in Figure 1, when 4 is using the obsolete route 4-3-1-0, 3 switch to the alternative routes fed by 4 and a forwarding loop forms. From this example, we can tell the root causes of forwarding loops are those ”ghost” routes. Accordingly, a good solution eliminating transient loops ought to be able to remove ”ghost” routes. In this sense, the existing BGP convergence acceleration solutions are good candidates. Due to page limitation, we will not elaborate these solution. Please refer to [4] and [5] for further reading.

B. *Solutions for transient routing failures*

The fundamental cause of transient routing failures is the limited alternative route visibility due to constraints from protocol itself and the network topologies. As shown in Figure 1, because 3 and 4 can inform their current best paths only, 1 leans no more than its best path learned from 0 directly. Once 0-1 fails, 1 has to experience transient routing failure before it learns alternative routes from 3. Therefore, the solutions that ameliorate the transient routing failures should improve the alternative route visibility during the failover moment.

1) Multi-path re-routing: BGP is a single route routing protocol, i.e. it allows nodes exchange their best routes only, which restrict the visibility of the alternative routes. In order to mitigate the limitation, a straightforward solution is to allow nodes to inform their neighbors multiple routes instead of the best ones. When link failure happens, nodes can re-route through the additional alternative routes immediately. Thus the transient routing failures are avoided. For example, node 4 in Figure 1 inform 1 all of its routes. Although 1 discards the first two paths due to poison reverse and AS path loops, the third one 4-5-2-0 is kept. Once 0-1 fails, 1 will use alternative route 4-5-2-0 to forwarding packets and will not experience transient routing failure.

However, in order to ensure the routing protocol works correctly with the presence of both the best route and the alternative routes, a node, say u needs to differentiate its best route from its alternative routes when it sends these routes to its neighbors. Meanwhile, a neighbor of u, say v should not choose its best route from alternative routes from its neighbors given the presence of best routes from the neighbors. Otherwise, if v chooses a best route from u's alternative routes but u is still using its best route, a routing consistency is resulted and the forwarding loop forms. Accordingly, in the path selection procedure, the routes labeled as the relevant neighbors' alternative routes should be inferior to the normal routes.

Besides the potential forwarding loops that the multi-path re-routing scheme might be suffering, the scheme is also limited from the perspectives of follows.

At first, the scheme is too expensive to implement in terms of not only the memory and bandwidth consumption for the additional alternative routes, but also the complexity to modify the standard BGP and the compatibility issues.

Second, multi-path re-routing scheme actually cannot eliminate transient routing failure completely, depending on the number of paths that the routers are allow to exchange, which in turn is determined by the topologies and routing policies of the relevant nodes. For example, in Figure 1, except that routers are allowed to send at least 3 paths, 1 would not learn 4's third alternative route 4-5-2-0 to avoid transient routing failure. In order to eliminate transient failure completely, nodes would rather announce all paths in their rib_ins. However, this would further worsen the system overhead.

Finally, although the multi-path re-routing scheme might ameliorate transient routing failure, it might exacerbate transient forwarding loops. For example, in Figure 1, if 1 does not know 4's third alternative route, 1 will experience transient failure. But if 1 knows it, before 4 receive 1's route change and switch to 5-2-1, a forwarding loop between 1 and 4 forms when 1 use the alternative route 4-5-2-0.

2) Indicative re-routing: Based on the spirit of multi-path re-routing scheme, a much simpler re-routing scheme can be developed. In the multi-path scheme, no matter how many alternative routes a node is allow to send to its neighbors, the alternative routes are never be used until failures happen and the relevant nodes cannot find any other route than the alternative ones. Also, even if an alternative route is chosen as the best route, the usability of the route is still in question: the alternative route might be still an alternative one or even vanish after the convergence. Thus, the information that an alternative route conveys is rather limited – just an implicit hint of the potential exits to the destination. So, it makes no sense to let alternative routes consume the same resources as the ordinary routes while containing such limited, in most of time useless, information. In fact, the same amount of information can be fully expressed by a simple indicator of alternative routes piggybacked with each route. The intuition leads to a much simpler and more light-weight scheme – *indicative re-routing*.

In the indicative re-routing scheme, each route is tagged with an alternative route indicator, which indicates whether the sender has alternative routes besides the current best one. The details are described as follows.

At first, at the sender side, when the sender, say u, changes its best route and is ready to inform one of its neighbors, say v. u will examine whether the best route and other alternative routes in its rib_in can be sent to v, i.e. whether they are permitted by the protocol and u's export policies. If u's best route is the only route can be sent to v, the indicator piggybacked with the best route is set to $alternative\_none$; if besides the best route one of the alternative routes can also be sent to v, the indicator is $alternative_exists$; if none of them can be sent, u will send v a withdrawal. However, if the best route is not permitted but one of the alternative routes can be sent, u will generate a new route with empty AS path and set the indicator to $alternative$ and then send to v. This route is the delegation route of u's alternative routes that can be sent to v. We call the route whose indicator is $alternative$ an *indicative route*. Note that before an indicator route is sent out, it should also pass the relevant route export operations, e.g. u's AS number should be appended to its AS path if u and v are in different ASs.

On the other hand, at the receiver side, when v receives a route from u, if the route meets its import policies, the route will be installed in v's rib_in. However, if the route does not comply with v's routing policies but the route is tagged with an indicator of $alternative\_exists$, which implies that although u's current best path is unacceptable but u still has some other alternative routes available, v will install an indicative route as if it was an indicative route originated in u.

Accordingly, the path selection procedure needs to be revised. An indicative route is always inferior to ordinary routes. As a result, the indicative routes should not be used until no ordinary route is available. At the same time, since the indicative routes just represent the existence of alternative routes, there is no substantial difference between different indicative routes. Therefore, if only indicative routes left in a node's rib_in and one of them is installed as best route, even if some new indicative routes arrive, the path selection procedure will not be evoked to compare indicative routes if the current best one is still available. When an indicative route is selected as the best route, the node should send a withdrawal back to the sender of this indicative route as poison reverse. The node should further advertise its choice of indicative route to the other neighbors as if the chosen indicative route is a ordinary route. The indicative routes inform the relevant neighbors that there exist alternative routes in the directions they were sent. In the case that an node lost all its ordinary routes to the destination, it can install one of its indicative routes to forward packets. In this way, transient routing failures are eliminated.

Note that an indicative route actually replaces part of the role of withdrawal in the standard BGP. In order to accelerate the propagation of indicative routes during the failover moment, the indicative routes are excluded from the regulation of the MRAI timer. As a result, the indicative routes can be propagated as soon as possible. Therefore, even if a node temporarily empties its loc_rib, it takes at most $O(d)$ to catch an indicative routes. Therefore, the indicative re-routing scheme ensures that the duration of transient routing failures at a node is upper bounded by $O(d)$.

Finally, we use the example in Figure 1 to show how the indicative re-routing scheme works. Before the failure of link 0-1, 1 installs the best path from 0 and two indicative routes 3-? and 4-?. Once 1 detects the link failure, it has to choose one of the indicative routes, say 3-? as the best route. Then 1 sends a withdrawal to 3 and an indicative route 1-3-? to 4. When 3 receives the withdrawal, it will switch to 3-4-1-0. At this moment, 3 has no other route except the best route, which cannot be sent to 4. Thus 3 sends a withdrawal to 4 and route 3-4-1-0 with indicator of $alternative\_none$ to 1. 4 will receive 1's indicative route at almost the same time and switch to 4-3-1-0 before 3's withdrawal arrives and then sends an indicative route to 3 and 4-3-1-0 with indicator of $alternative\_exists$ to 1. After 1 receives 3's route, it detects an AS loop and discard the indicative route from 3 but choose that from 4. After 4 receives 3's withdrawal, 4 switch to the final route 4-5-2-0 and inform both 1 and 3 the valid route. Finally, all nodes converges to the new routes through 5.

Compare with the multi-path scheme, the indicative re-routing scheme is much simpler and more light-weight. It consumes almost the same amount of resource to store and send routes as the standard BGP but needs only slight modification of standard BGP. However, like the multi-path scheme, the indicative re-routing scheme also introduce excessive transient forwarding loops. For the example in Figure 1, before node 4 finally switch to path 4-5-2-0, there is a forwarding loop involving 1, 3 and 4.

C. Solutions for transient failures and loops

As described before, the ghost-flushing or EPIC schemes are good at removing obsolete routes and eliminate transient forwarding loops while the indicative re-routing scheme accelerates the propagation of alternative routes and mitigates transient routing failures. By combining the two types of schemes, we achieve a solution that can eliminate both transient loops and failures. As an example, we describe the combined indicative+EPIC scheme as follows. Similarly, we can combine the indicative re-routing and the ghost-flushing to achieve the same goal.

Because the operation of the indicative re-routing and the EPIC are quite orthogonal – the former manipulates the indicator and the latter operates the $fesnList$, in most cases, the two schemes can operate independently without any inference. Special attention needs to be made in the following two situations. At first, during the convergence process, when a node is removing the obsolete routes according to the receive $fesnList$, an removed route with indicator of $alternative\_exists$ should be replaced by an indicative route as if this indicative route is received from the relevant neighbor. On the other hand, when an indicative route is generated at either the sender or receiver side, if an AS number is appended to the AS path in the indicative route, the corresponding $fesnList$ should be updated. We will examine the performance of the indicative+EPIC scheme with simulations.

TABLE I
SIMULATION TOPOLOGIES

#AS	29	110	208	409
#router	44	208	443	1001
#router reflector	0	14	40	104
#eBGP session	118	572	1188	2692
#iBGP session	34	220	550	1490

IV. EXPERIMENTAL EVALUATION OF SOLUTIONS

In this section, we use simulations to evaluate the performance of the solutions. We implement an event-driven simulator simBGP [17], which supports the standard BGP, the BGP convergence acceleration schemes, i.e. ghost-flushing and EPIC, the multi-path and the indicative re-routing schemes, and the indicative+EPIC re-routing scheme.

A. Simulation settings

1) Timer settings: In order to distinct the difference between the inevitable one-hop routing delay d and the timer-induced delay D, we artificially set d extremely small by setting the processing delay at each node uniformly distributed in $[0.001, 0.01]$ millisecond and the queuing delay at each link uniformly distributed in $[0.01, 0.1]$ milliseconds and the link bandwidth 100MB. In this way, the routing delay d is almost 0. The MRAI timer for eBGP sessions is set to 30 seconds and for iBGP sessions 5 seconds. In real BGP implementations, the MRAI timer is peer-based. In other words, all the announcements to any destination sent from a node to one of its peers are spaced out by the same timer. In our simulation, there is only one destination. In order to mimic the behavior of the peer-based timer, which might be triggered by the announcements to other destinations, we set the MRAI timer in the following way. When a node is ready to send routing messages to one of its peer, if the MRAI timer for this peer is not set, the message will be blocked for a duration uniformly distributed between $[0, MRAI]$. That is, the MRAI timer is assumed to have been set by the background announcements.

2) Topology settings: The simulations are performed based on the AS level topologies provided by BJ Premore [16]. In order to obtain the more realistic BGP system topologies, i.e. a router level BGP system composed of routers peering with both iBGP and eBGP sessions, we extends these AS level topologies into router level. At first, we split a node peering with more than 4 neighbors into several router nodes, which are fully connected via iBGP sessions while the peers are randomly assigned to one of them and connected via eBGP sessions. If an AS owns too many routers, we further group these routers into clusters and assign a route reflector to each cluster to create a 2-tiered iBGP system. In this way, we produces the router level BGP systems with various peering structures, as shown in Table I.

3) Simulation scenarios: In simulation, we pick one AS to originate and announce the destination prefix. When every router reaches stable states, we break one of the eBGP links of the origin AS. Because ASs in the simulation topologies are densely connected, the link failure will trigger a failover event. We repeat the operation for every eBGP link of every AS 5 times and examine the average value of the relevant metrics. We simulate the scenarios where the following schemes are employed: the standard BGP (BGP), the ghost-flushing (ghost), the EPIC (EPIC), the multi-path re-routing scheme that allows maximal 2 paths exchanging between nodes (Multi-path 2), the indicative re-routing scheme (indicative), and the combined indicative+EPIC scheme (indicative+EPIC).

B. Evaluation Metrics

In the simulation, we use the sum of the duration of transient forwarding failures and forwarding loops that all the nodes experiences for the purpose of evaluation. The duration of transient failures or loops on a node is calculated in the following way. In the simulation, once a node changes its best path, the simulator will check whether it experiences transient forwarding failure or loops by constructing its forwarding path at the moment. If it does, then all nodes whose forwarding paths contain this node will suffer the same transient failures or loops. Finally, we record the starting and ending time of the transient failures or loops and get the duration of transient failures and loops at every node.

Suppose that during the simulation, every node sends packets to the destination with a constant rate M packets per second. If the sum of the duration of transient failures and loops of a simulation scenario is H. The total packet loss due to the transient forwarding failures and loops is $H \times M$. Therefore, the sum of duration of the transient failures or loops is actually equivalent to the end-to-end packet loss during the failover events.

At the same time, we also record the number of messages generated during the failover convergence process and the convergence time.

C. Simulation Results

Figure 2(a) shows the performance of different schemes in various topology settings. The X axis indicates the simulation topology. For instance, ”net 29” represents the topology composed of 29 ASs. Three values are recorded for each routing scheme in each topology: the sum of the duration of transient forwarding loops, the sum of the duration of transient forwarding failures, and the sum of both, which are represented by the heights of three bars. Two shoulder-by-shoulder narrows bars, depicting the loops on the left and the failures on the right, is over the wide bar, which represents the total duration of failures and loops.

At the same time, Figure 2(b) and (c) shows the convergence time and the message numbers for the different routing schemes and topologies.

From Figure 2, our observations are summarized as follows.

1) standard BGP: During the failover convergence process, transient failures and loops coexist. With the increase of the topology size, transient loops becomes more prominent than the transient failures, because the visibility of alternative routes is increasing with the increase of the topology size.

2) Ghost-flushing and EPIC: Although the ghost-flushing and EPIC eliminates the transient forwarding loops, they exacerbates the transient forwarding failures. Compared with the standard BGP, they even worsen the total duration of transient failure and loops, which means they even lengthen the duration of end-to-end reachability disruption.

3) Multi-path and indicative re-routing schemes: The re-routing schemes do eliminate the transient forwarding failures. But they literally transform the transient failures to loops. So in terms of the total duration of failures and loops, the re-routing schemes make no obvious progress. Compared with multi-path scheme, indicative re-routing scheme generates fewer messages and converges a little faster. At the same time, multi-path scheme that limits no more than 2 paths exchanging between nodes cannot eliminate transient forwarding loops completely.

4) indicative+EPIC scheme: As expected, the combined scheme can not only eliminate transient forwarding failures but also transient forwarding loops. However, the cost is that it generates more routing messages than any other schemes.

Finally, although none of the schemes can accelerate the convergence speed, the indicative+EPIC scheme does eliminate the transient forwarding failures and loops and improve the end-to-end reachability during the failover convergence process.

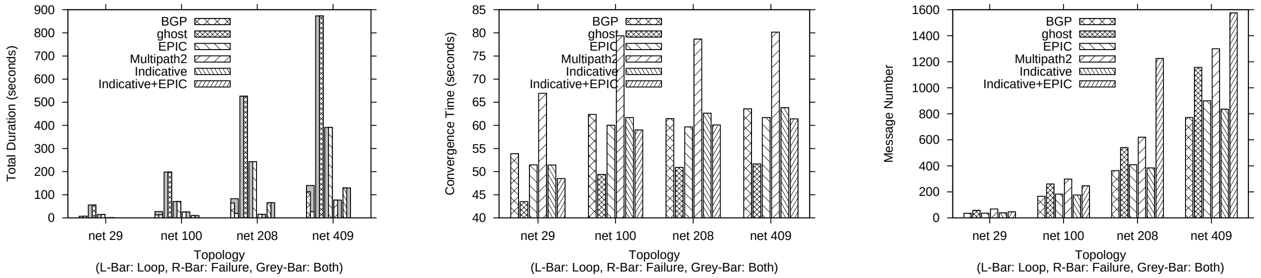
V. Related work

Previous studies have shown that degraded end-to-end path performance is correlated with routing dynamics [1], [6], [8]–[10], [19]. Recent study has shown that a significant number of transient routing failures occur during route convergence [21]. Our work focus on how to achieve fast re-routing during route convergence.

Significant works have been done to achieve fast rerouting for IGP. Recent work shows sub-second IGP convergence time can be achieved by fine tuning parameters of IGP [20]. Our work focus on fast rerouting during BGP convergence. MPLS based approaches use pre-computed backup paths to reroute around failures immediately after detecting link failures. However, this method is usually done in a centralized manner [13].

Nelakuditi *et al* [12] use interface specific forwarding tables to achieve fast re-routing, which requires a new algorithm for forwarding table calculation. Narvaez *et al* [11] focus on link state protocols and propose a local restoration algorithm. Bonaventure *et al* [3] propose to establish redundant protection peering sessions through tunnels to feed alternative routes in case of failures. On the contrary, our approach focuses on the modification of BGP protocol.

Wang *et al* in [22] have proposed to store alternate paths in routing table so that a particular destination has two entries, one corresponding to the normal next-hop on the shortest path



(a) Duration of transient failures and loops

(b) Convergence Time

(c) Message Number

Fig. 2. Simulation results

and other corresponding to the alternative next-hop. However, this approach doubles the routing table size and cannot be applied to BGP. Our approach increases the availability of route but does not increase the routing table size.

VI. Conclusion

In this paper, we propose solutions for BGP to eliminate the transient failures and loops during the failover convergence process. At first, we identifies two major causes of transient end-to-end reachability disruptions, the transient failures and transient loops. On the one hand, we find that the existing BGP convergence acceleration solutions, such as the ghost-flushing or EPIC, can eliminate transient loops. However, they exaggerate transient failures. On the other hand, we introduce an indicative re-routing scheme to eliminate transient routing failures. Nonetheless, the solution works with the expense that the transient forwarding loops are exacerbated. Finally, we reach a combined indicative+EPIC scheme, which exploits the advantages of the both types of schemes and eliminates not only transient failures but also loops. We use simulations to examine the performance of the solutions.

Acknowledgment

The authors are supported by the NSF grant CNS-0325868, ANI-0208116 and ANI-0085848, and Alfred P. Sloan Fellowship.

References

[1] S. Agarwal, C.-N. Chuah, S. Bhattacharyya, and C. Diot. The Impact of BGP Dynamics on Intra-Domain Traffic. In *Proceedings of ACM SIGMETRICS*, 2004.

[2] A. Basu, C.-H. L. Ong, A. Rasala, F. Shepherd, and G. Wilfong. Route Oscillations in I-BGP with Route Reflection. In *Proceedings of ACM SIGCOMM*, Pittsburgh, PA, USA, August 2002.

[3] O. Bonaventure, C. Filsfils, and P. Francois. Achieving Sub-50 Milliseconds Recovery Upon BGP Peering Link Failures. In *Proceedings of the 2005 ACM CoNEXT*, pages 31–42, Toulouse, France, 2005.

[4] A. Bremler-Barr, Y. Afek, and S. Schwarz. Improved BGP Convergence via Ghost Flushing. In *Proceedings of IEEE INFOCOM*, March 2003.

[5] J. Chandrashekar, Z. Duan, Z.-L. Zhang, and J. Krasky. Limiting Path Exploration in BGP. In *Proceedings of IEEE INFOCOM*, 2005.

[6] N. Feasmster, D. Andersen, H. Balakrishnan, and M. Kaashoek. Measuring the Effects of Internet Path Faults on Reactive Routing. In *Proceedings of ACM SIGMETRICS*, San Diego, CA, USA, June 2003.

[7] U. Hengartner, S. Moon, R. Mortier, and C. Diot. Detection and Analysis of Routing Loops in Packet Traces. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurement*, Marseille, France, 2002.

[8] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet routing convergence. *IEEE/ACM Transactions on Networking*, 9(3):293–306, June 2001.

[9] C. Labovitz, A. Ahuja, and F. Jahanian. Experimental Study of Internet Stability and Wide-area Network Failures. In *Proceedings of Fault Tolerant Computing Symposium*, June 1999.

[10] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.-N. Chuah, and C. Diot. Characterization of Failures in an IP Backbone. In *Proceedings of IEEE INFOCOM*, Hong Kong, China, March 2004.

[11] P. Narvaez, K. Siu, and H. Y. Tzeng. Local Restoration Algorithm for Link-state Routing Protocols. In *ICCCN*, 1999.

[12] S. Nelakuditi, S. Lee, Y. Yu, and Z. Z.L. Failure Insensitive Routing for Ensuring Service Availability. In *IWQoS*, 2003.

[13] P. Pan, G. Swallow, and A. Atlas. Fast Reroute Extensions to RSVP-TE for LSP Tunnels. Internet draft, IETF, February 2004.

[14] V. Paxson. End-to-End Routing Behavior in the Internet. *IEEE/ACM Transactions on Networking*, 5(5):601–615, October 1997.

[15] D. Pei, X. Zhao, D. Massey, and L. Zhang. A Study of BGP Path Vector Route Looping Behavior. In *Proceedings of the 24th ICDCS*, pages 720–729, Tokyo, Japan, March 2004.

[16] B. Premore. Multi-AS topologies from BGP routing tables. http://www.ssfnet.org/Exchange/gallery/asgraph/index.html.

[17] J. Qiu. simBGP, a simple BGP simulator. http://www.bgpvista.com/simbgp.php.

[18] Y. Rekhter, T. Li, and S. H. Ed. A Border Gateway Protocol 4 (BGP-4). RFC 4271, IETF, January 2006.

[19] M. Roughan, T. Griffin, Z. M. Mao, A. Greenberg, and B. Freeman. Combining Routing and Traffic Data for Detection of IP Forwarding Anomalies. In *Proceedings of ACM SIGCOMM NeTs Workshop*, Portland, OR, USA, August 2004.

[20] A. Shaikh and A. Greenberg. OSPF Monitoring: Architecture, Design and Deployment Experience. In *Proceedings of USENIX Symposium on Networked Systems Design and Implementation*, 2004.

[21] F. Wang, L. Gao, J. Wang, and J. Qiu. On Understanding of Transient Interdomain Routing Failures. In *Proceedings of IEEE ICNP*, Boston, MA, USA, November 2005.

[22] Z. Wang and J. Crowcroft. Shortest Path First with Emergency Exits. In *Proceedings of ACM SIGCOMM*, September 1990.