Fast Failure Recovery Using Multi-threading in BGP

Gao Lei and Lai Mingche

Department of Computer,
National University of Defense Technology, Changsha 410073, China
mingchelai@nudt.edu.cn

Abstract. Since Border Gateway Protocol presents poor path diversity at the router level, it does not allow for a fast recovery and always produces a great amount of BGP churn upon session failures, making inter-domain routing system facing reliability challenge. In this paper, we propose a failure recovery scheme implemented on threaded BGP architecture to quickly react and recover from session failures. In our scheme, a failure recovery thread is designed to work with multiple BGP session threads in parallel, and it employs BFD protocol to quickly detect link failures and generate backup routes by our presented dynamic route generation algorithm in case of session failure occurrence. The failure recovery thread will notify BGP session threads to establish backup paths with related nexthops along the paths, and it will not advertise route change to all neighbors as traditional BGP does until the notification timer expires, thus effectively reducing BGP churn and improving the route stability. Finally, simulation results show that the execution time of dynamic route generation algorithm is only 160ms averagely, the increased number of update messages relative to BGP is 16.4% and 48.8% at most under two experiment scenarios, and the duration of packet loss reaches 0.82s and 2.63s on average compared to 6.5s of BGP, effectively improving the reliability of BGP.

Keywords: Failure recovery, TBGP, Multi-threading, Reliability.

1 Introduction

With more and more real-time network applications participating in the Internet, BGP being the default domain routing protocol plays an important role in the end user's perception of network performance. Despite of the remarkable availability and responsiveness demonstrated by the Internet in most cases, they still need a substantial improvement when BGP session failures strike. The reliability of Internet depends on the reaction time necessary for the underlying routing protocols finding the backup paths in case of failures[3]. It was found that 82% of the failures lasted less than 180 seconds[1], and the recovery time was longer than 20s for 17% of the failure because these long recovery time was attributed to the failure detection time and the time that distributed the BGP messages during the BGP convergence[14]. ISPs typically use the convergence of the routing protocol to quickly react to the session failures by advertising route withdrawn and update messages, but the convergence time was always in the order of a few or a few tens of seconds[1]. Furthermore, Labovitz[15]

James J. (Jong Hyuk) Park et al. (eds.), *EMC Technology and Service*,
LNEE 181, pp. 163–170, DOI: 10.1007/978-94-007-5075-3_19,

showed that the convergence delay for isolated route withdrawals could take more than 3 minutes in 30% of the cases, and Zhao[16] showed that packet loss rate can increase by 30x and packet delay by 4x during recovery. The above facts reveal that the conventional routing protocols cannot satisfy the requirement of mission-critical applications which are sensitive to routing changes, and even not provide with efficient resilience and stability when facing the failures.

In this paper, we propose a failure recovery scheme which is implemented on TBGP architecture[20] and works as an independent failure recovery thread in parallel with BGP session threads. The failure recovery thread as the critical component of the scheme employs Bidirectional Forwarding Detection(BFD)[21] protocol to quickly detect link failures and generate backup routes by our presented dynamic route generation algorithm in case of session failure occurrence. The simulation results show that the average execution time of the dynamic route generation algorithm is about 160ms, the increased number of BGP messages relative to tradition BGP is respectively 16.4% and 48.8% at most, and the duration of packet loss reaches 0.82s and 2.63s on average compared to 6.5s of BGP.

2 Related Works

Many literatures have studied the impact of BGP routing failures to network and put forward various methods for fast network failure recovery in order to avoid disruptions. Several earlier works [4,5,6] have made efforts to analyze the impact of BGP session failures by analytical modeling and testbed experiments. Recent works[1-3,7-13,17-19] mostly emphasized on exploiting routing diversity and used the multipath mechanism to provide a primary path and multiple backup paths for fast failures recovery. Bonaventure[1] proposed a fast reroute technique to quickly react to inter-domain link failures and update the FIB by using pre-established IP tunnel, but it could not ensure continuous connectivity in case of multiple failures. [12] provided alternated path on forwarding plane by establishing IP tunnel in case of failures, but it selectively advertised tunnel routes to the nexthop so as to decrease global routing table size. Wang[9] defined BGP routing planes to indicate multiple BGP routes and especially proposed a crank-back technique, allowing to perform traffic diversion in case the failed link didn’t have any feasible alternate route. Wei[3] proposed a pure IP-based fast failover method which defined primary backbone protocol to propagate the primary route, such as OSPF, IS-IS or EIGRP, and backup backbone protocol like BGP. When failures happened on the primary route, routers would switch to backup route before receiving failure notification. [2] proposed the main Add-Paths selection modes to advertise to iBGP peers some primary received paths so as to provide route diversity. [7] employed similar method with [2] and it could reduce the control plane convergence time and almost eliminate the failure recovery time because no other AS than the provider BGP learned about the failures, but it also increased the message number and the control plane stress that need compute and store the received backup paths. Other methods like [8] exploited the additional potential routing diversity by relaxing BGP peering links and setting up BGP sessions between Internet exchange points, aiming to overcome the inherent constraint of the existing BGP-compliant recovery schemes.

3 Our Proposal

3.1 Basic Idea

The failure recovery thread is designed in our failure recovery scheme by utilizing the mechanism of multi-threading, to execute with BGP session process threads in parallel. The work of the failure recovery thread mainly includes three aspects as follows. Firstly, it repeatedly detects the states of all neighbor sessions and discovers the failures by using BFD protocol, which is now widely adopted in the backbone routers to detect link failures in the order of milliseconds. Secondly, it calculates backup routes for each destination node that the router could reach. Usually, the router may have one or multiple routes to a destination router in routing table, and only the optimal route is advertised to neighbors. The failure recovery thread will determine or generate backup routes to cope with BGP session failures. Thirdly, it decides when to use backup routes according the link state information collected by BFD and updates the routing table and forwarding table. Once the failure recovery thread discovers a session failure, it immediately checks the backup routes and finds an available alternate route, and then it updates the forwarding table to prevent packet loss. Similarly, it also checks the optimal routes to other destinations in the routing table and uses the backup routes instead of the routes that include the failed links.

3.2 Dynamic Route Generation Method

The objective of dynamic route generation method is to compute the backup routes so that no packets are discarded upon the occurrence of session failures. This method is used when no backup route is available in case of link failure and its main idea is to detour the failed links and find out in the routing table the alternative routes to nexthop along the primary route. The method detail is depicted in algorithm 1. Since the session connecting nexthop router r_f is failed and no backup route is available for reaching destination node r_s in initial, the algorithm will find feasible path to following routers along the failed path $r_f, r_{f+1}, r_{f+2}, \ldots, r_s (s>f)$ from r_f in routing table. Then if algorithm finds one or multiple paths $p_1, p_2, .., p_t (t \geq 1)$ to the midterm router $r_i (f \leq i < s)$, and these paths follow the relation of $p_1 \geq p_2 \geq \ldots \geq p_t$ where $p_j \geq p_k$ means p_j is superior to p_k on distance, it will choose an available path which is with relatively shorter distance value and does not include failed links belong to failed link set $\{l_1, l_2, \ldots, l_m\}$ dynamically detected by the failure recovery thread. Thus in line 4 the selected path p_u is combined with the right part of the primary route from r_i, and also added into the routing table as a backup route. If there is no path to the current router, algorithm will be iterative and continue to find paths to the next router along the failed primary path. But if algorithm finds none feasible path to arbitrary midterm node along the primary route, the reasons may be that the network encounters multiple session failures so that many feasible paths are blocked, and then algorithm will choose the neighbor router with the largest packet traffic as the nexthop, which is neither the upstream neighbor nor the failed one, so as to prevent packet loss.

Initial:

(1)The failed path is $r_f,r_{f+1},r_{f+2},\ldots,r_s(s>f)$, and the failed link is the session to nexthop r_f;

(2)There is no backup route to destination r_s;

(3)The set of failed links is $\{l_1,l_2,\ldots,l_m\}$.

Algorithm 1: Dynamic route generation algorithm

```
1  Pick next node $r_i(f\leq i<s)$ in path $r_f,r_{f+1},r_{f+2},\ldots,r_s$, and find in routing table if
   there is path to $r_i$, and $i$ begins with $f$;
2  If there are one or multiple paths $p_1,p_2,..,p_t(t\geq 1)$ to $r_i$ and these paths follow
   the relation of $p_1\geq p_2\geq\ldots\geq p_t$ where $p_j\geq p_k$ means $p_j$ is superior to $p_k$ on distance,
   goto 3; else if $i$ is not $s$-1, goto 1, else goto 6 ;
3  Check $p_1,p_2,..,p_t$ in turn to find a path that does not include failed link belonging
   to $\{ l_1,l_2,\ldots,l_m\}$ and has the smallest path suffix;
4  If finding such a path $p_u$, i.e. $n_1, n_2, \ldots, r_i$, combine the path $p_u$ with the
   right part of the primary route from $r_i$, i.e. $n_1, n_2, \ldots, r_i, r_{i+1},\ldots,r_s$, and add
   the new path to the routing table as a backup route, goto 7, else goto 5;
5  If $r_i$ is not $r_{s-1}$, go to 1; else goto 6;
6  Select a neighbor router with the largest traffic as the nexthop, which is not
   the upstream neighbor or the failed one, and then goto 7;
7  Algorithm finishes.
```

3.3 The Failure Recovery Scheme Implementation

The modified TBGP architecture is shown in Fig. 1. All BGP sessions are distributed to process on multiple BGP session threads, and one failure recovery thread is built to discover session failure using BFD protocol and recover failure as soon as possible.

The failure recovery thread is created after the construction of routing table by master thread of TBGP, and it charges for collecting the states of BGP sessions and generating the backup routes. As there may be multiple feasible paths for reaching some destinations, the optimal path is always the primary route, and other paths are prepared for backup routes. Thus the failure recovery thread first travels throughout the routing table to tag the backup routes which often prefer to choose second-optimal routes. And for those destinations with unique path, the failure recovery thread will compute a backup route with the dynamic route generation algorithm. When the failure recovery thread detects a session failure, it checks the routing table to finds the affected optimal routes by the failure. For these primary routes destructed by failure, the thread first detects the usability of their backup routes which could not include failed links, and then determines whether to use backup routes or generate new backup route by the dynamic route generation algorithm. After that, the failure recovery thread temporarily cancels primary route and tags backup route as current optimal route in routing table and also modifies the nexthop in forwarding table according to backup routes. Then it notifies the route changes to each BGP session thread. With this method, the traffic could be quickly diverted to safe path as encountering link failures, which ensures the continuous transfer and prevents packet loss.

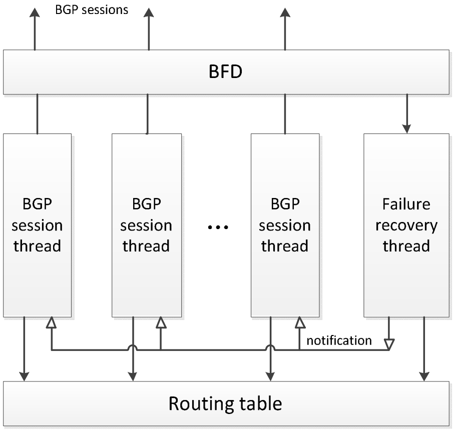


Fig. 1. Modified TBGP architecture

As we know, many failures are short-lived and recover quickly. Thus, when the failure recovery thread receives the notification of BGP session recovery from BFD, it resumes the primary routes in routing table and also updates nexthops in the forwarding table. But still some failures could not recover in short period as we imagine, a notification timer in the failure recovery thread is set to determine whether and when to advertise route update or withdrawn messages to neighbors. If the failure recovers before notification timer expires, router will not advertise route update messages. Otherwise, the thread will notify every BGP session thread the route changes as notification timer expires as traditional BGP does upon failure occurrence. The delayed route change advertising by this method could help with reducing BGP churn and improving the route stability. On the other hand, when BGP session thread receives the route change notification from the failure recovery thread, it only advertises new routes to those routers that are nexthops in backup routes. And it needs to reestablish session in case that the failed session recovers again.

4 Simulation and Evaluation

We simulate on dual quad-core Xeon server with Linux 2.6.18-8AX the efficiency of dynamic route generation algorithm and evaluate the performance of our scheme in terms of the number of BGP updates generated during session failures and the duration of packet loss. In detail, we use Ghitle[22] to generate AS-level topologies that include the business relationships between ASes, and define the internal structure of each AS and the way that ASes interconnect on the router-level with its neighbor.

4.1 Dynamic Route Generation Algorithm Efficiency

In this experiment, we use AX4000 series to construct network environment and generate session failures with the failure proportion of 10%. A mass of routes generated by AX4000 are injected into network with the number from one million to ten millions, and these routes are configured to different destinations. Then the statistic average execution time by VTune^{TM} is shown in Fig.2. From the results, we could find that the average execution time of the algorithm rises with the scale of routing table, and it is about 160 milliseconds on average in all configurations. The reason for the linear increase of average execution time is that the algorithm need travel throughout the routing table and find feasible paths that exclude failed links, more route entries and failed links mean larger traveling time and path comparing time.

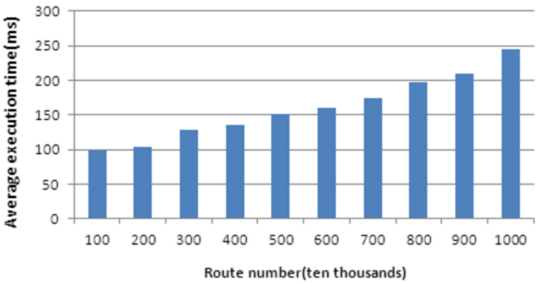
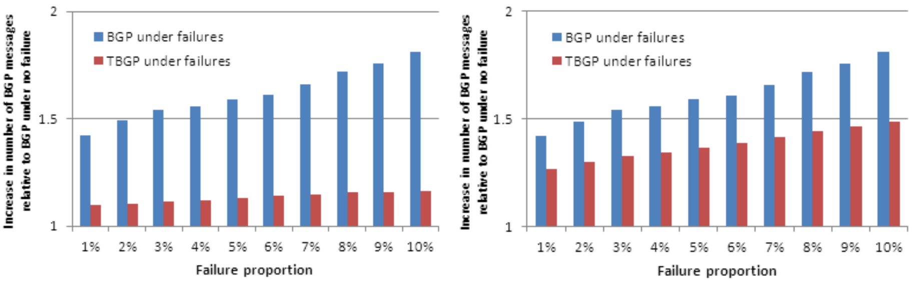


Fig. 2. Average execution time of dynamic route generation time

4.2 Increase on BGP Update Message

To evaluate the increased number of BGP updates during session failures, the experiment topology is composed of 8-10 ASes and each AS owns 30-50 sessions by tool Ghitle. And when constructing routing table, each router has 20 percent routes redundant with others which are primary routes in routing table. We randomly configure a set of failures which take up 10% of total session links at most. Then we set two simulation scenarios. The scenario I is that all failures recover before the notification timer in the failure recovery thread expires and the period of the timer is set to be 30 seconds. Since BGP session threads in TBGP need not advertise update message throughout the whole network before the notification timer expires, the increase of update messages in Fig.3(a) is relatively small, and the increase is only 9.8% to 16.4% with the failure proportion increment. The increased messages are mainly those advertised to nexthops indicated by backup routes. However, as BGP advertises lots of update and withdrawn messages during session failures, the number of BGP update messages increases greatly, reaching 42% to 81%. In scenario II, we make 40% of the failures recover after the notification timer expires. With the failure proportion increases in Fig.3(b), the increase in number of BGP update messages by TBGP goes up to 48.8%, which increases obviously in comparison with scenario I. That's because BGP session thread has to advertise lots of update messages to neighbors as traditional BGP does. The increased messages are mainly BGP update messages, but the total increase is still much fewer than that of traditional BGP as encountering failures.



(a) (b)

Fig. 3. Increase in number of BGP messages relative to BGP under scenario I(a) and II(b)

4.3 Duration Time of Packet Loss

With the simulation scenarios in section 4.2, we further consider the duration time of packet loss shown in Table 1. The average duration of packet loss with traditional BGP is about 6.5 seconds, and that of our scheme with BFD is about 0.82s under scenario I and 2.63s under scenario II on average.

Table 1. Average duration time of packet loss(s)

	Max value	Min value	Average
BGP under failures	8.34	4.23	6.5
TBGP in scenario I	1.27	0.46	0.82
TBGP in scenario II	3.58	1.86	2.63

5 Conclusion

The scheme proposed in this paper employs BFD protocol to quickly detect link failures and generate backup routes by our presented dynamic route generation algorithm in case of session failure occurrence. Simulation results show that our scheme greatly reduces the number of BGP messages and the duration of packet loss, efficiently improving the reliability of BGP.

Acknowledgements. This work was supported by NSFC (No. 60803153, No. 61103188 and No. 61070199) and the National High Technology Research and Development Program of China (No. 2012AA01A301).

References

1. Bonaventure, O., Filsfils, C., Francois, P.: Achieving sub-50 milliseconds recovery upon bgp peering link failure. IEEE/ACM Transactions on Networking 15(5), 1123–1135 (2007)

2. Schrieck, V.V., Francois, P., Bonaventure, O.: BGP add- paths: the scaling/performance tradeoffs. IEEE Journal on Selected Areas in Communications 28(8), 1299–1307 (2010)
3. Wei, Z., Wang, F.: Achieving resilient routing through redistributing routing protocols. In: ICC, pp.1–5 (2011)
4. Xiao, L., He, G., Nahrstedt, K.: BGP Session Lifetime Modeling in Congested Networks. Journal of Computer Networks 50(17), 1–12 (2006)
5. Wang, L., Saranu, M., Gottlieb, J.M., Pei, D.: Understanding bgp session failures in a large isp. In: IEEE INFOCOM 2007, pp. 348–356 (2007)
6. Sahoo, A., Kant, K., Mohapatra, P.: Characterization of BGP Recovery Time under Large-Scale Failures. In: IEEE ICC 2006, pp. 949–954 (2006)
7. Wang, H., Wang, C., Cai, S.: Enhance Internet Routing Availability with Multipath Inter-domain Routing. In: IEEE ICCDA 2010, vol. 5, pp. 428–433 (2010)
8. Hu, C., Chen, K., Chen, Y., Liu, B.: Evaluating Potential Routing Diversity for Internet Failure Recovery. In: IEEE INFOCOM 2010 (2010)
9. Wang, N., Guo, Y., Ho, K., et al.: Fast Network Failure Recovery Using Multiple BGP Routing Planes. In: IEEE GLOBECOM 2009 (2009)
10. Ragha, L.L., Chag, K.V.: Multiple Route Selector BGP (MRS-BGP). In: IEEE ICWET 2010, pp. 304–308 (2010)
11. Dai, B., He, J., Wang, H., et al.: iRoute: A Scalable Inter-domain Multi-path Routing Framework for Multimedia Transmission. In: IEEE ICMT, pp. 4966–4969 (2011)
12. Ma, H., Zhang, J., Guo, Y., He, L.: Scalable Resilient BGP – Fast Recovery from Transient Inter-Domain Link Failures. In: IEEE IITA, pp. 980–984 (2008)
13. Watari, M., Hei, Y., Ano, S., Yamazaki, K.: OSPF-based Fast Reroute for BGP Link Failures. In: IEEE GLOBECOM, pp. 1–7 (2009)
14. Pei, D., der Merwe, J.V.: BGP convergence in virtual private networks. Presented at the Internet Measurement Conf., Rio de Janeiro, Brazil (October 2006)
15. Labovitz, C., Ahuja, et al.: Delayed internet routing convergence. In: Proc. ACM SIGCOMM 2000, Stockholm, Sweden, August 28-September 1, pp. 175–187 (2000)
16. Zhao, X., Pei, D., Massey, D., Zhang, L.: A study on the routing convergence of Latin American networks. In: Proc. LANC 2003, LaPaz, Bolivia, October 4-5, pp. 35–43 (2003)
17. Kushman, N., Kandula, S., Katabi, D., Maggs, B.M.: RBGP: Staying connected in a connected world. In: Proc. NSDI, pp. 341–354 (2007)
18. Wang, F., Gao, L.: A backup route aware routing protocol –Fast recovery from transient routing failures. In: Proc. INFOCOM 2008 (2008)
19. Ganichev, I., Dai, B., Godfrey, P.B., Shenker, S.: Yamr: Yet another multipath routing protocol. EECS Department, Tech. Rep. UCB/EECS-2009-150 (October 2009)
20. Wang, K., Gao, L., Lai, M.: A scalable multithreaded BGP architecture for next generation router. In: IEEE EMC 2011, pp. 72–77 (2011)
21. Katz, D., Ward, D.: Bidirectional forwarding detection. RFC 5880 (July 2010)
22. Delaunois, C.: Ghitle: Generator of Hierarchical Internet Topologies using LEvels, http://ghitle.info.ucl.ac.be/