Cluster Comput (2017) 20:817–833
DOI 10.1007/s10586-017-0771-1

A BGP session takeover method for high availability based on virtualization technique

Hwasung Kim[1] · **Sangil Kim**[1] · **Hoyong Ryu**[2] · **JaeHyung Park**[3] · **Jinsul Kim**[3]

Received: 10 November 2016 / Accepted: 30 January 2017 / Published online: 3 March 2017

Abstract Recently, the improvement in the data communication networks in terms of availability and reliability is emerging as a critical issue. In this context, the high availability and reliability of routers that control data flow and routing path between networks are being recognized as critical problems. Many methods for providing high availability to minimize the loss cost caused by the failure of a router are being studied. This paper analyzed the takeover method proposed in the previous studies to support the high availability of the BGP protocol, and proposed a BGP session takeover method using the virtualization technique to improve the takeover performance.

Keywords BGP · High availability · Takeover · Virtualization

✉ Hwasung Kim
hwkim@kw.ac.kr

Sangil Kim
rlatkd2342@kw.ac.kr

Hoyong Ryu
hyryu3@etri.re.kr

JaeHyung Park
hyeoung4@chonnam.ac.kr

Jinsul Kim
jsworld5@chonnam.ac.kr

[1] Department of Electronics and Communications Engineering, KwangWoon University, Seoul, Republic of Korea

[2] Network S/W Platform Research Team, ETRI, Daejeon, Republic of Korea

[3] School of Electronics and Computer Engineering, Chonnam National University, Gwangju, Republic of Korea

1 Introduction

The rapid development of Internet has made it possible to handle a variety of tasks everywhere in the world only with a web browser. Not only has the world been connected through a big loop called Internet, but also our daily businesses such as Internet banking, e-payment, e-commerce, Home Shopping, security transactions, and civil affairs have been conducted using Internet. Likewise, as the Internet has penetrated deep into our lives, and the number of Internet service users has grown rapidly, the demand for uninterrupted service has gradually increased.

In particular, the high availability and reliability of the router to control the data flow and routing path between networks in a data communication network has been recognized as a very important issue. In order to ensure the high availability and reliability, a non-stop active routing technology has been studied as a technique for maintaining a session so that the client and the peer can continue to communicate without recognition of the communication disconnection although the actual communication between the client and the peer is disconnected.

In the routers, routing processes such as BGP, LDP, and OSPF are executed, and these processes perform actual routing control functions such as mutual information transmission and routing table management. In case of failure on the router, the TCP connection, which belongs to transport layers, is disconnected due to the protocol characteristics; and thus, routing protocols using TCP such as BGP cannot maintain connection any longer. Especially in the case of BGP, even if failure is repaired and operation resumes, the contents of routing tables need to be exchanged with peer routers after session is re-established, which results in long down time. That is, it is required to support not only the redundancy of TCP connection, but also the redundancy of the BGP for the non-stop active routing.

The technologies associated with the non-stop active routing include ‘FT-TCP’, which allows a faulty server to keep its TCP connections open until it either recovers or it is switched over to a backup [1], and ‘application-driven TCP recovery and non-stop BGP’ that proceed with the takeover by separating the application takeover and the TCP layer takeover [2]. In the case of ‘FT-TCP’, the TCP is responsible for all takeovers of TCP-based applications, which places a huge burden on the TCP layer. The ‘application-driven TCP recovery and non-stop BGP’ technology also poses a problem in that although the overall BGP takeover time was reduced by duplicating the BGP data, it still need to re-establish the TCP connection.

In addition, the existing high availability (HA) redundancy methods, to which the non-stop active routing technology is applied, have mainly dealt with the redundancy in the physical hardware environment. If the HA redundancy technology is applied in the physical hardware environment, a high cost arises due to the purchase of the HA support equipment. Therefore, in this paper, a container technology, which is a type of virtualization technology, was used to achieve cost savings and reduce BGP takeover latency time.

The organization of this paper is as follows. Section 2 investigates high availability system-related technologies such as ‘FT-TCP’ and ‘application-driven TCP recovery and non-stop BGP’, and Sect. 3 defines the backup data for BGP session takeover. Then, Sect. 4 describes a BGP session takeover method not applying the virtualization technology, and a BGP takeover method applying the virtualization technology. Section 5 proceeds with simulation performance evaluation and Sect. 6 present the conclusions.

2 Background

2.1 High availability system

The high availability system implies ability to protect application from failure in CPU (central processing unit), hard disk, or network component and to ensure the continuous service of mission critical application and operating environment. That is, in case of service interruption, service restoration should be made possible within the shortest time.

High availability is largely divided into non-stop active routing, graceful restart, and Layer 2 availability. The non-stop active routing technology is a functional element of high availability system that provides seamless routing by means of routing engine redundancy in order to enable the client and a peer to communicate each other continuously and to maintain sessions without being aware of communication disconnection even if the communication disconnection has actually occurred between the client and the peer. The routing engine redundancy uses two routing engines of ‘active’ and ‘standby,’ and switches to the standby routing engine in case of failure in the active routing engine, without changing the adjacent neighbor router path.

The graceful restart is a method for supporting redundancy to network equipment through collaboration with adjacent routers. The Layer 2 availability support routing engine redundancy by transmitting Layer 2 control information to the backup engine in the same way as the routing engine redundancy [1].

2.2 FT-TCP (fault-tolerant TCP)

In ‘FT-TCP’, the TCP server fault tolerance is provided with the use of standardized backup [2]. Two kinds of wrappers (South Side Wrapper and North Side Wrapper) located around the TCP forward TCP byte stream to a logger and ensure that the TCP state stored in the logger is consistent with the active TCP state. In the case of a TCP failure, a standby TCP is operated by means of replaying using the state stored in the logger. During this failover period, the client’s TCP connection is continuously maintained by sending zero-size window advertisements at regular intervals.

2.3 Application-driven TCP recovery and non-stop BGP

The paper on the ‘application-driven TCP recovery and non-stop BGP’ introduces an application-driven approach. The TCPR introduced in the paper is a middle box, which is not different from a NAT box. The application driven recovery approach introduced in the paper maintains seamless TCP connection with peer router by splitting the TCP connection using the TCPR in the re-connection process in case of failure, and resolves the re-connection of the application session using backup. This idea helps to avoid a large burden of taking the responsibility for the application state in the TCP layer. In addition, the effects of a simple form of the application-driven recovery, which is lightweight and easy to implement, can be obtained by assigning check pointing responsibilities on the connection state along with other states to the application [3].

2.4 BGP performance comparison between BGP re-connection method without redundancy and redundancy method

In this subsection, a simulation that compares the throughput and recovery latency time of the BGP re-connection process to which redundancy is not applied and BGP takeover method with redundancy applying the previously studied method [2,4] was carried out to confirm the performance of the redundancy method.

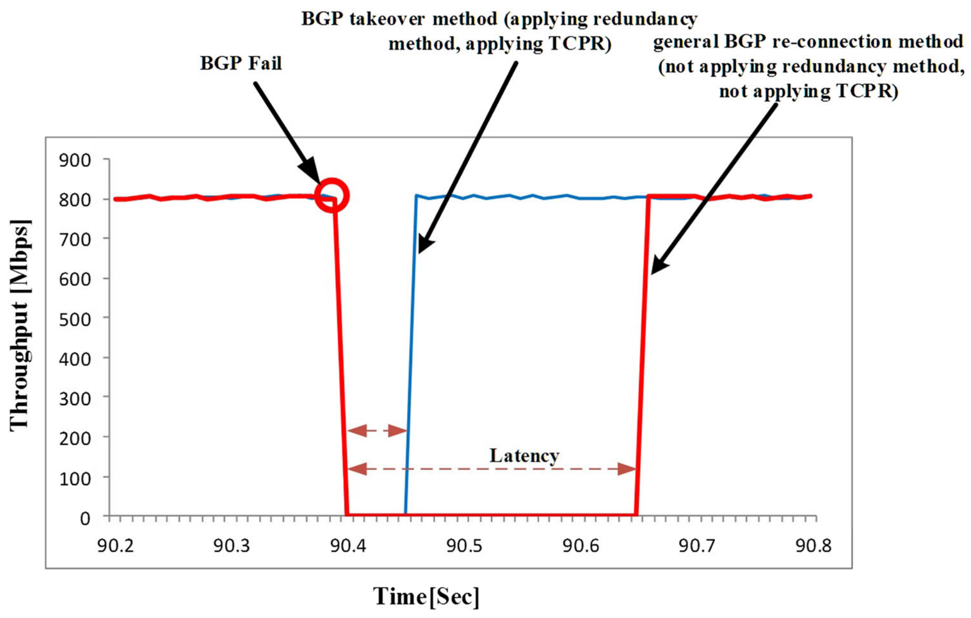


Fig. 1 Comparison of BGP throughput and latency time (hold time = 0 s)

Generally, Hold time has great effects on BGP re-connection, so we carried out a simulation to distinguish how long BGP re-connection process took actually except the fault detection time, by setting hold time to 0 s. So, the disconnection of BGP session may be detected at the same time point of failure. Fig. 1 shows the simulation results when hold time is 0 s. In Fig. 1, the average throughput of the redundancy method was 686.73 Mbps, and that of the BGP re-connection method not applying the redundancy method was 631.24 Mbps, which indicates that the BGP takeover method with redundancy showed better performance with 8.1% improvement in throughput.

The reason why the redundancy method showed better performance is as follows. In the case of BGP re-connection method without redundancy, the throughput is reduced because of much packet loss during the BGP and TCP re-connection process. The BGP takeover method with redundancy shows better throughput because BGP takeover latency time is much shorter than BGP re-connection latency time due to the BGP data backup, even though it also has some packet loss during the takeover process.

The latency time was also shorter in the case of BGP takeover method with redundancy than BGP re-connection process not applying the redundancy method. In the case of redundancy method, the takeover time related to the TCP was 0.06 s, and the takeover time related to the BGP was 0.07 s, showing a total of 0.13 s latency time. It took 0.42 s in case of BGP re-connection method.

Figure 2 shows the simulation results when hold time was set to 50 s. When hold time = 50 s, the average throughput of the redundancy method was 683.56 Mbps, and that of the BGP re-connection method was 275.21 Mbps, which indicates that the BGP takeover method showed much better performance with a 60.4% improvement.

The BGP re-connection method not applying the redundancy begins BGP re-connection process after the hold time is expired, because BGP recognizes the disconnection of BGP session only when BGP does not receive any keepalive message within hold time interval. Therefore, the latency increases as much as the hold time. On the other hand, redundancy method detects the disconnection of BGP session earlier, because standby BGP detects an active BGP failure using the heartbeat message without waiting for the expiration of hold time. Therefore, redundancy method showed the shorter latency.

Figures 1 and 2 showed the simulation results using 100 RIB entries, not considering the general size of BGP RIB. Typically, the number of entries in BGP RIB is about 100–150 thousands, Fig. 3 shows the simulation results using 100 thousand entries in case of hold time = 50 s. In Fig. 3, the average throughput of the BGP takeover method with redundancy was 622.56 Mbps, and that of the BGP re-connection method not applying the redundancy method was 85.21 Mbps, indicating that the BGP takeover method showed better performance with 89.3% improvement.

In the case of the BGP re-connection process not applying the redundancy method, there exists a process where the RIB table entries are updated. Therefore, the larger the size of the RIB table entries, the greater the value of the latency. That is, as identified from the simulation results, since not only the hold time of 50 s but also the time required for the process of updating the RIB table entries are included, much clearer difference in performance is displayed, compared to the previous results.

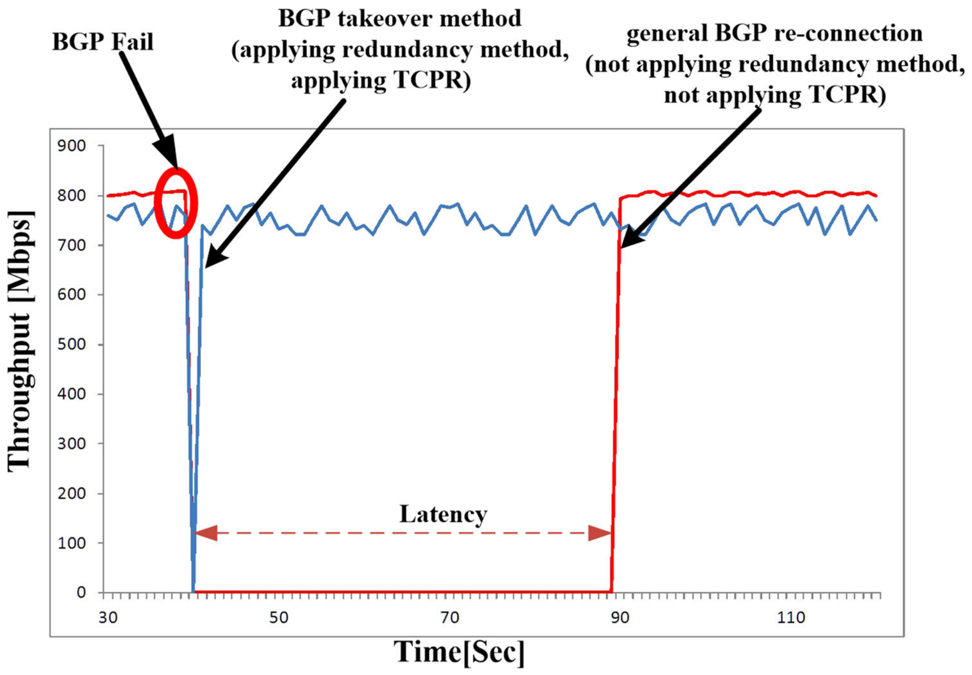


Fig. 2 Comparison of BGP throughput and latency time (hold time = 50 s)

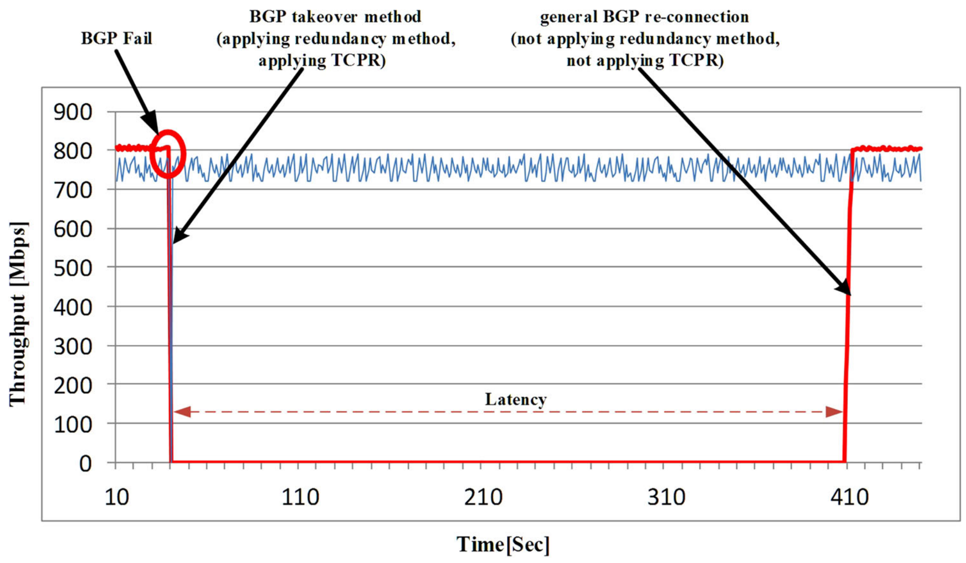


Fig. 3 Comparison of BGP throughput and latency time (hold time = 50 s, 100 thousands entries)

The simulation results summarized in Table 1 revealed that the BGP takeover method with redundancy showed better performance in terms of the average throughput, and latency time mainly due to the much shorter BGP takeover latency. However, both methods have the TCP latency. Even in the case of BGP takeover method with redundancy, a new TCP connection should be established between standby router and TCPR.

That is, in order to improve the performance of the BGP takeover method with redundancy, it is required to reduce the BGP takeover latency time further, and to decrease the latency time that occurs in the TCP layer. In this regard, this paper proposed the BGP takeover method with higher performance resolved by applying the virtualization technology to reduce both of the BGP takeover latency and the latency that occurs in the TCP layer.

Table 1 Comparison of BGP average throughput and latency time

	BGP takeover method applying the Redundancy method	BGP re-connection method not applying the redundancy method
Throughput (hold time = 0 s)	686.73 Mbps (100%)	631.24 Mbps(91.9%)
Throughput (hold time = 50 s)	683.56 Mbps (100%)	275.21 Mbps(40.2%)
Throughput (RIB = 100 thousand)	622.56 Mbps (100%)	85.21 Mbps (11.7%)
Latency (hold time - 0 s)	0.13 s	0.42 s
Latency (hold time = 50 s)	0.13 s	50.52 s
Latency (RIB = 100 thousand)	0.13 s	395.52 s

3 Backup data definition and backup points for BGP session takeover

3.1 Definition of backup data for BGP session takeover

Requirements for BGP session takeover are as follows:

- Which data of the active router should be stored on the standby router for the sake of BGP session takeover?
- At which (time) points should data of the active router be stored on the standby router for the sake of BGP session takeover?

For BGP session takeover, session related information delivered by BGP messages and other BGP operational information such as RIB (Routing Information Base) in the active router should be stored in the standby router.

Among BGP messages, messages having the session related information of a BGP session are the BGP Open message and the BGP Update message. BGP Open message, which is used for setting a BGP session after TCP connection has been established between BGP neighbors, contains ASN (Autonomous System Number), hold time, destination address, and path information. And the BGP Open message sent by the other contains destination address information and keepalive time information to maintain a BGP session. Thus, if the standby router is to substitute the active router, the received BGP Open message must be stored.

The BGP Update message has Path Attribute items, in which path information is stored, and contains the address information and path information of destination; and thus the backup of BGP Update messages sent and received is essential. While the BGP Open message has to be backed up only once during the session because it is a message exchanged only at the time of BGP session establishment, the Update message need to be stored for each sent and received message, because the Update message is sent whenever BGP path information changes. Figure 4 shows backup data to be backed up for BGP session takeover.

There are eight BGP Path Attributes in the BGP Update message, and the path attributes fall into one of ‘Well-known

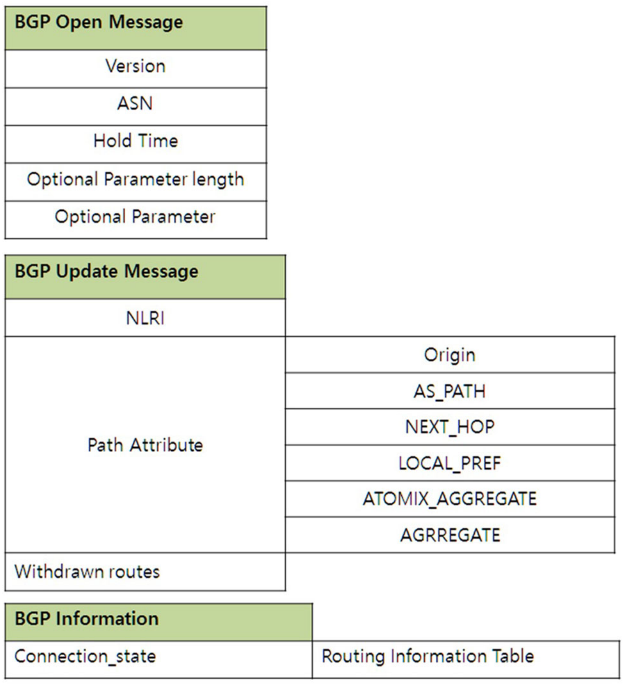


Fig. 4 Defined BGP data elements for backup

Mandatory’, ‘Well-known Discretionary’, ‘Optional Transitive’, and ‘Optional Non-transitive’.

The ‘Well-known Mandatory’ includes attributes that must be included in the BGP Update message. It must be supported in BGP implementation; and in case of its absence, notification error will occur. The ‘Well-known Discretionary’ refers to elements that are recognized by all BGP vendors, but do not cause notification error even if not included in BGP Update messages.

The ‘Optional Transitive’ may or may not be implemented by every BGP vendor, and BGP passes its relevant value on to other AS because it is transitive, even if BGP does not understand the relevant attribute value. The ‘Optional Non-transitive’ may or may not be implemented by every BGP vendor, like the ‘Optional Transitive’; however, the relevant information should not be sent to BGP peers regardless of

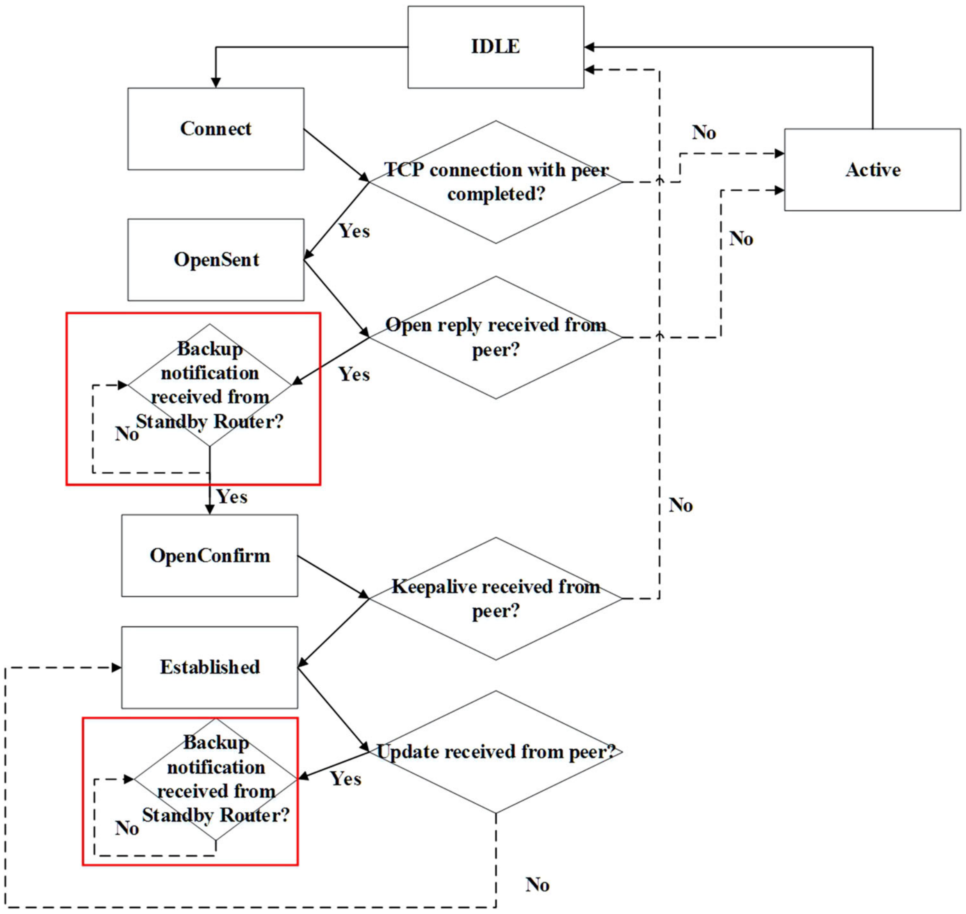

IDLE
Connect
TCP connection with peer completed?
No
Active
Yes
OpenSent
No
Open reply received from peer?
Backup notification received from Standby Router?
Yes
No
Yes
No
OpenConfirm
Keepalive received from peer?
Established
Update received from peer?
Backup notification received from Standby Router?
Yes
No
No

Fig. 5 Backup points for BGP takeover

whether the BGP router understands the relevant attribute or not, for it is non-intransitive [5].

Therefore, among the eight BGP path attributes existing in the BGP Update message, attributes falling into the 'Well-known Mandatory' and the 'Well-known Discretionary' should be stored, and the relevant data elements are defined in Fig. 4. Additionally, Connection_state data, which is information about the operation state of the current active router, and RIB held by the active router should be backed up together to the standby router.

3.2 Backup points for BGP session takeover

As shown in Fig. 5, one of the backup points for BGP data is when BGP transitions from the OpenSent State, in which an Open message is sent, to the OpenConfirm State. And if the Open message has been received normally from the peer, the fields of the Open messages and the Connection_state information defined in Fig. 4 are stored from the active router to the standby router.

The other backup points for BGP data is when BGP is in Established State. In the Established State, whenever an Update message is sent and received, the fields of the Update message defined in Fig. 4, Connection_state information, and RIB information need to be stored from the active router to the standby router.

In addition, as for the point of time when backup data are stored for BGP session redundancy, the backup points should be defined considering TCP connection; states before the establishment of TCP connection and the states after the establishment of TCP connection. The states before the establishment of TCP connection include the Idle State and the Connect State. Because the Idle State and the Connect State are the states that configure the TCP connection before the BGP connection, in these states there is no BGP data to backup. Therefore for these two states, only the TCP related data, not BGP data, needs to be backed up, and in this paper TCP related data means TCPR data because the previously researched TCPR was used in this paper [2].

The states after TCP connection establishment include the OpenSent State, the OpenConfirm State, and the Established State. A message sent in the OpenConfirm State is a BGP Keepalive message. The backup of keepalive message is not required, because the keepalive message is a BGP message header with no data. Thus, only the Connection_state information, which is information on the current state of active router, is stored when receiving the keepalive message.

4 A method of BGP takeover for high availability

This section describes two BGP takeover methods; First method to which a virtualization technology is not applied and second method to which a virtualization technology is applied. The existing BGP takeover method is implemented in the legacy equipment, where the virtualization technology is not applied, and it works in an environment where the active and standby routers operate on different physical nodes. If they operate on different physical nodes, the same physical equipment as the active router to be used as a standby router is needed.

This results in a problem of cost. In addition, the latency problem arises because of copying the RIB table is required, and the TCP needs to be re-connected. In this regard, this paper sought to investigate the BGP takeover method based on the virtualization technology. Towards this end, a Docker container technology [6] was used to create active and standby containers respectively on a physical node. Using the virtualization, the additional physical node is not required for the standby, leading to a reduction in cost, and an issue on the TCP re-connection also disappears since the network stack of the kernel can be shared by each container due to the characteristics of the Docker container.

In addition, the RIB table used in the active can be used directly in the standby through shared data volumes, and as a BGP daemon that operates in the active container can be easily taken over to the standby container only with the configuration information for BGP settings, the BGP takeover time can be further reduced, as compared to the existing method. The details of each method are described below.

4.1 BGP takeover method not applying a virtualization technology

BGP takeover method not applying a virtualization technology was used as the redundancy based BGP takeover method for the simulation described in Sect. 2.

If BGP fails because an error occurs in the active BGP, the active router notifies the standby router of the error occurrence, and the standby router starts BGP, using BGP backup information. This functionality was implemented in RM (Redundancy Module) for the simulation. RM is a function module that is implemented in both of active and standby routers. RM performs the synchronization functions such as backup and takeover between active and standby routers.

As for the backup operation, while BGP is running, the active RM captures BGP messages and stores those to the standby RM. Then, standby RM transmits ACK for the backup data message to the active RM to acknowledge successful receipt of the information.

As for the takeover, the RM of the active router checks periodically failure in BGP that operates on the active router; and in case of a failure in BGP, the active RM notifies the failure to the standby RM at once so that BGP takeover can occur.

Figure 6 shows the sequence of BGP takeover method not applying a virtualization technology. When standby RM detects the failure of active BGP, BGP takeover process is started by standby RM. It first starts the BGP daemon, then configures BGP and TCPR using backed up data. Then it re-establishes a TCP connection between new active BGP and TCPR.

4.2 BGP takeover method applying a virtualization technology

The BGP takeover method applying a virtualization technology uses a Docker technology. Docker refers to a container-type virtualization technology based on LXC (Linux Containers) as a new container-based virtualization tool that appeared in 2013.

It has very similar features to those of a VM (Virtual Machine), but it can be implemented in a much lighter form than that of VM. In the case of VM, it is installed on the top of Hypervisor (VMware, KVM, Xen, etc.) that performs the virtualization of hardware. The VM is considered to be a kind of virtualized execution environment. Various kinds of OS such as Linux or Windows can be installed as a host or a guest OS.

On the other hand, Docker engine provides the container as a virtualized execution environment by making the virtualization of the host OS, not virtualization of hardware like VM. Docker engine, which is running between host OS and container, can be used only on Linux family OS as a host OS. Also, only the Linux family OS is allowed as a guest OS in container. However, the guest OS image does not include the whole duplicated kernel image of host OS. guest OS uses the host OS kernel as it is, and only different parts between host OS and guest OS are packaged within the container. For example, if host OS is Ubuntu version x, and guest OS is CentOS version y, only the different parts between Ubuntu version x and CentOS version y are packaged in the container, not the full image of CentOS version y. This is the reason why container is lighter than VM.

Since Docker stores data of container to the data volume managed by host OS, not the container area, the data volume can be shared between the containers. The data volume is accessed directly by the containers rather than through the File System to reduce the access time.

As far as the networking is concerned, when the docker is installed, a virtual network interface called docker0 is cre-

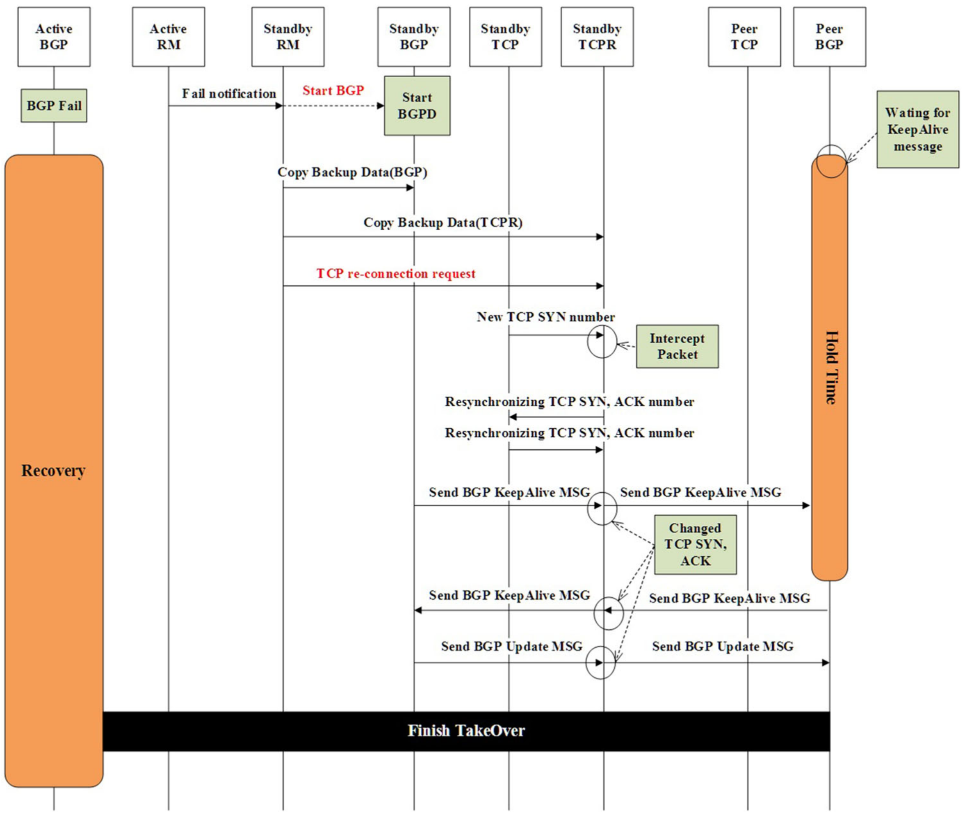


Fig. 6 Sequence of BGP takeover method not applying a virtualization technology

ated for the host as shown in Fig. 7. The docker0 is a virtual Linux bridge for the container to communicate with other containers. The bridge is basically based on L2 communication, and if the container is created, another virtual network interface called eth0 is also created for the container by Linux namespace and it is bound to bridge docker0 to make the peer interface. Thus, when the container is communicated to the outside, the traffic is delivered through eth0 and docker0 [6].

This paper proposed a BGP takeover method based on the Docker technology described above. As shown in Fig. 7, two containers are created based on a BGP daemon image in one physical node. Of the two containers, one is operated as active router, and the other as standby router. The active router backs up and stores its own BGP backup data such as BGPd.conf and RIB to the data volume when BGP update message is exchanged or BGP settings are changed by an administrator. The configuration information for eth0 interface, which was set to the active router, is also stored to data volume in addition to the BGP backup data. The backup data stored in the data volume is shared with the standby.

The active router communicates with peer router through the protocol stack of the host OS via the docker0 bridge. TCP connection information is stored in the host OS that is shared among containers. So, the backup of TCP connection information is not required.

If the active router fails, the standby router starts a BGP daemon in the standby router container through the Docker run command using the stored BGP configuration in shared data volume. Since the RIB table is provided through shared data volumes, it is used as it is, and with the use of eth0 interface information backed up from the active router, the standby router eth0 interface is configured equally to the active router to constitute the connection with the Docker0 Bridge.

As shown in Fig. 8, BGP takeover latency time ($\boldsymbol{T}Latency$) is composed of failure detection time($\boldsymbol{T}Detection$) and BGP configuration setup time($\boldsymbol{T}BGPtakeover$) in standby router. Formula 1 indicates the total latency time of the BGP takeover method applying a virtualization.

$$\mathbf{T}Latency = \mathbf{T}Detection + \mathbf{T}BGPtakeover \tag{1}$$

The failure detection is performed by Docker. A BGP session can be maintained by simply setting the BGP configuration. Also, in this case, the re-establishment of TCP connection is not required.

Figure 9 represents the latency time that occurs in the BGP takeover method not applying the virtualization technology. In this case, BGP takeover latency time($\boldsymbol{T}$Latency) is composed of failure detection time($\boldsymbol{T}$Detection) and BGP backup data copy and configuration setup time ($\boldsymbol{T}$cBGP), TCPR backup data copy and configuration setup time and

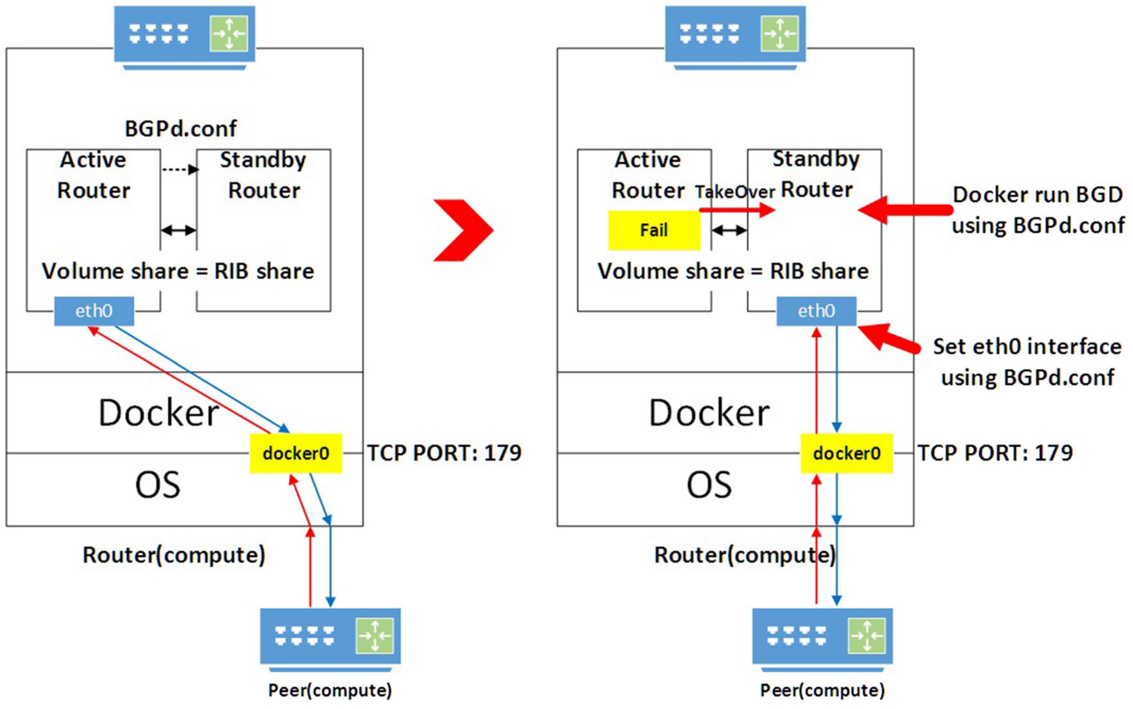
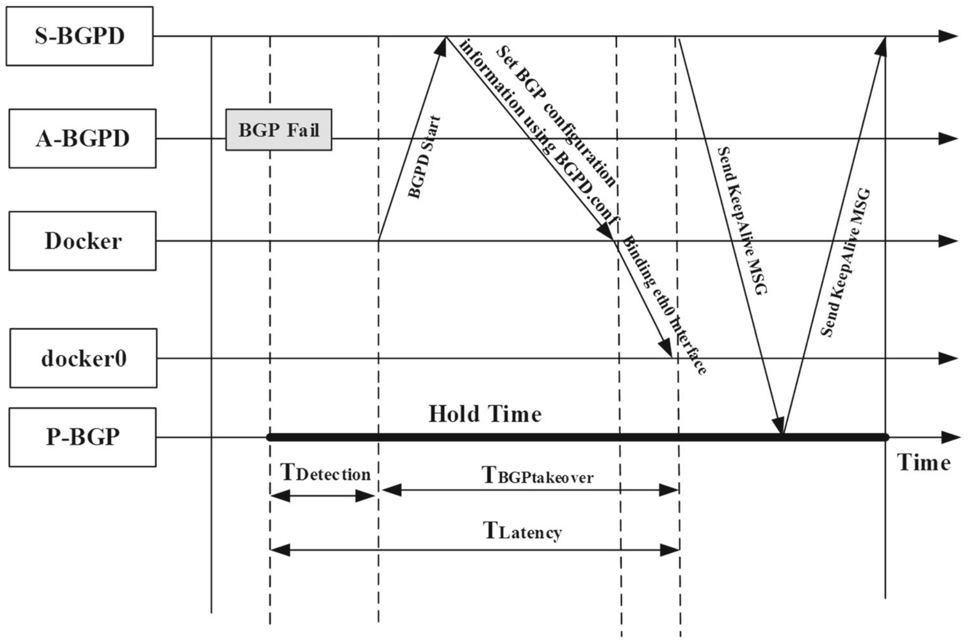


Fig. 7 BGP takeover method applying a virtualization technology

Fig. 8 Latency of BGP takeover method applying a virtualization



TCP connection switchover time(TcTCP), in standby router. Formula 2 indicates the total latency time of the BGP takeover method not applying a virtualization.

$$TLatency = TDetection + TcBGP + TcTCP \quad (2)$$

As shown in Figs. 8 and 9, the takeover latency time($TLatency$) should be less than hold time, which means that BGP takeover should be completed entirely within the hold time($TLatency < \boldsymbol{HoldTime}$), and if the hold time is expired before the takeover is completed, the peer recognizes that the BGP session is disconnected, and then reconnection of the BGP session and TCP connection is required. In the case of a BGP takeover method applying a virtualization, it does not copy the RIB table. So, if the hold time value is not set lower than the $TLatency$ value, a stable BGP takeover is possible.

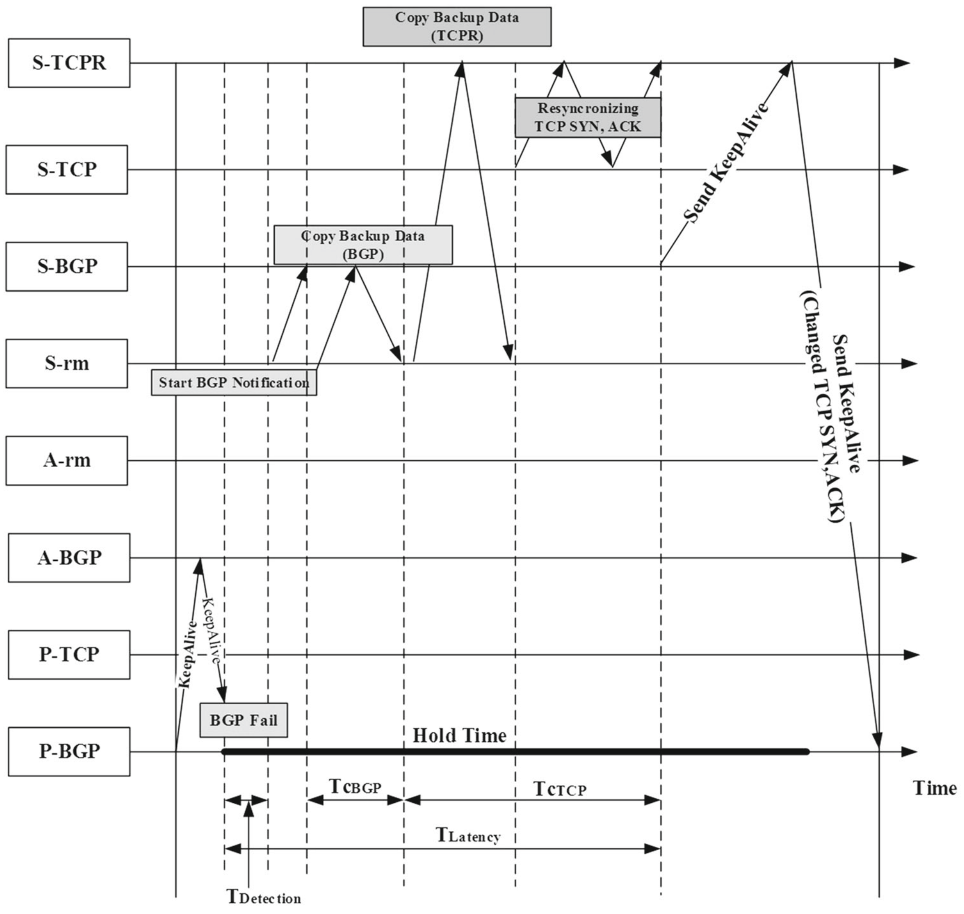


Fig. 9 Latency of BGP takeover method not applying a virtualization

However, in the case of a BGP takeover method not applying a virtualization, the ***TcBGP*** value, which is the time that configures the BGP and copies the RIB table, is determined in accordance with the size of the RIB table, so the ***TcBGP*** value carries a constraint condition like that of formula 3 in order to achieve a stable BGP takeover.

$$\mathbf{T}cBGP < \text{HoldTime} - (\mathbf{T}Detection + \mathbf{T}cTCP) \quad (3)$$

Therefore, the ***Maximum RIB Entries*** for achieving a stable BGP takeover can be calculated through formula 4.

$$\textbf{Maximum RIB Entries} = \mathbf{T}cBGP/\textbf{1 RIB Entry Copy Time} \quad (4)$$

5 Results of simulation

In this Section, simulation on the BGP takeover method applying the virtualization technology, the BGP takeover method not applying the virtualization technology described in Sect. 4, and the GR (Graceful Restart) was carried out. For performance analysis, the performance of GR was measured by installing exbpg [7] in a physical server, and that of BGP takeover method not applying virtualization technology was measured by installing exbpg in two physical servers. And the performance of the BGP takeover method applying the virtualization technology was measured by installing the Docker image in which exbgp was installed in a physical server with the same performance as that of the physical server used previously.

5.1 Comparison of BGP throughput

This section describes the performance evaluation results on the BGP throughput of the three methods. The BGP throughput evaluation was performed under the condition of hold time = 0 s, 50 or 100 thousand RIB table entries.

First, the throughput comparison under the condition of hold time = 0 s is shown in Fig. 10 and Table 2. The average throughput of the takeover method not applying the virtualization was about 686.73 Mbps, the takeover method applying the virtualization was about 795.32 Mbps and the GR was 746.35 Mbps. The performance evaluation results showed that the GR provided a better performance than the method not applying the virtualization, and the method applying the virtualization exhibited a superior performance to the other two methods. The

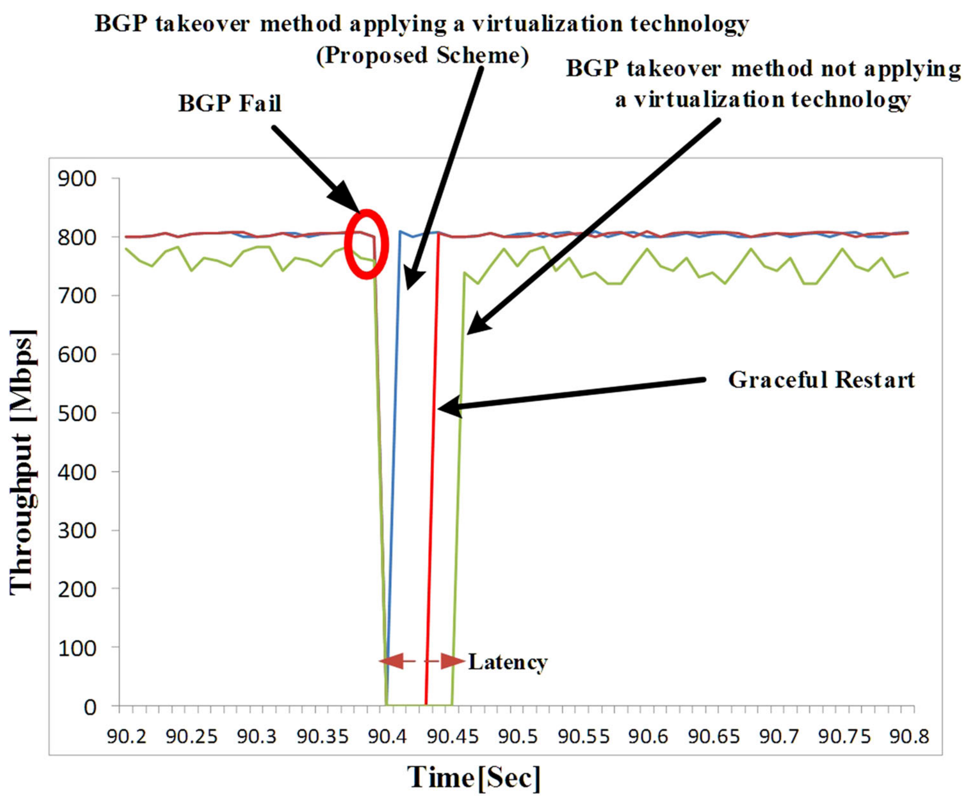


Fig. 10 Comparison of BGP throughput and latency time (RIB table entry = 50)

Table 2 Comparison of BGP average throughput & latency

	BGP takeover method applying a virtualization technology	BGP takeover method not applying a virtualization technology	Graceful restart
Throughput (RIB = 50)	795.32 Mbps (100%)	686.73 Mbps (86%)	746.35 Mbps (94%)
Throughput (RIB = 100 thousand)	782.71 Mbps (100%)	622.56 Mbps (83%)	732.48 Mbps (93%)
Latency (RIB = 50)	0.01 s	0.13 s	0.06 s
Latency (RIB = 100 thousand)	0.01 s	0.14 s	0.07 s

reason for the difference in the performance is as follows.

The method not applying the virtualization eliminates the BGP re-connection process by backing up the BGP data, however, it needs to copy RIB table of active BGP to standby BGP, which is a reason that it showed the poor performance than the other methods. In the case of GR, when the failure occurs in BGP, the actual data transmission is still maintained due to the separation of control plane and data plane, the BGP session re-connection is performed only in the control plane. The takeover method applying the virtualization outperforms the other methods, because the data volume for RIB table is shared and network stack is also shared between active BGP and standby BGP.

In the case of 100 thousand RIB table entries of Fig. 11, the simulation result showed the more performance difference between the two BGP takeover methods and GR than hold time = 0 s condition due to the RIB update time. In the case of the method not applying the virtualization in the simulation, it can show a better performance if active RIB table can be copied to standby RIB table before the takeover. Section 5.4 describes the performance evaluation results of modified RM that copies the active RIB table to standby RIB table in advance.

In the case of the BGP takeover to which the virtualization is applied, as the active and standby share the data volumes in which the RIB table is stored, it is not affected by the number of RIB entries.

In the case of GR, when the failure occurs in BGP, good performance can be provided because the actual data transmission is still maintained due to the separation of control plane and data plane, the BGP session re-connection is performed only in the control plane.

GR updates the RIB table in case of the recovery which affects the whole throughput of GR. However, in general, it shows better performance than the method not applying

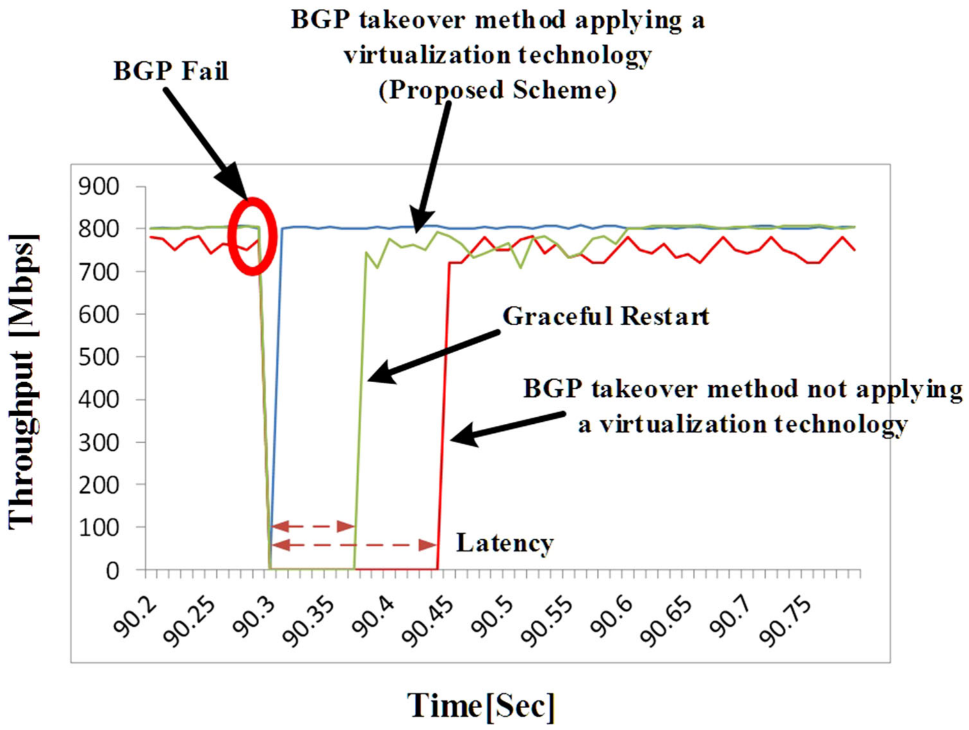


Fig. 11 Comparison of BGP throughput and latency (RIB table entry = 100 thousand)

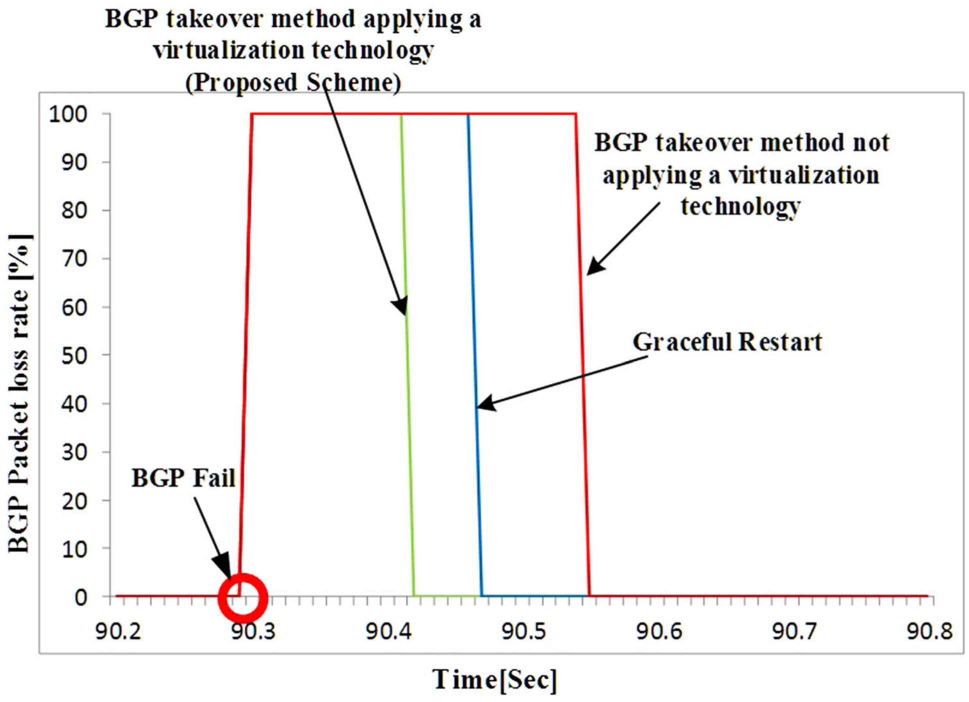


Fig. 12 Comparison of BGP packet loss by time

the virtualization in the simulation which always conducts backup before the takeover.

5.2 Comparison of BGP packet loss

Figure 12 shows the BGP packet loss due to the takeover or recovery. In order to compare the BGP packet loss during the BGP session takeover or session recovery, the hold time was set to 0 s, and as a result of a simulation, it is found that the BGP takeover method not applying the virtualization shows a more packet loss than other methods since it takes a longer takeover time.

In the case of BGP takeover method not applying the virtualization, there is no TCP segment loss because TCPR guarantees the continuity of TCP sessions. However, the BGP session should be switched over to standby router, so

Table 3 Comparison of BGP packet loss rate

	BGP takeover method applying a visualization technology (%)	BGP takeover method not applying a visualization technology (%)	Graceful restart (%)
Packetloss rate (1RIB Entry)	2.3	11.6	1.6
Packetloss rate (100 thousands entries)	2.6	17.2	1.6
Packetloss rate (over 1,168,762 entries)	2.8	87	1.9

BGP packet loss must occur. BGP takeover method applying the virtualization showed the less packet loss because its takeover latency is shorter than that of BGP takeover method not applying the virtualization.

Table 3 shows the BGP packet loss rate in which the number of RIB table entries is taken into account.

As described in Sect. 4, the BGP takeover method, not applying the virtualization technology, has a problem in that if the time needed for RM to copy the RIB table ($\boldsymbol{T}cBGP$) exceeds '***HoldTime*** - ($\boldsymbol{T}Detection$+$\boldsymbol{T}cTCP$)' the takeover cannot be performed within the hold time. In this paper, the calculated $\boldsymbol{T}cBGP$ value is 179.9894 s when the default hold time of 180 s, measured $\boldsymbol{T}Detection$ 0.005 s and measured $\boldsymbol{T}cTCP$ 0.0056 are applied to formula 3. And the result of substituting the $\boldsymbol{T}cBGP$ value to formula 4 draws out a ***Maximum RIB Entry*** value of 1,168,762 entries. If RIB has more than 1,168,762 entries, even the BGP takeover method not applying the virtualization should proceed with the BGP reconnection. However, the other two methods are not affected by the number of RIB entries.

5.3 CPU time consumption

This section describes the performance evaluation results on the CPU time consumption of the three methods. Figure 13 shows the CPU time consumption value for each method in case of BGP takeover. First of all, while the average CPU time consumption value of BGP takeover methods not applying the virtualization is 50%, and it is 72% on average especially in case of takeover. The BGP takeover method not applying the virtualization consumes more CPU time during the takeover than that of other methods because the data backup from active RM to standby RM occurs before the takeover (Fig. 13).

While the average CPU utilization value is 30% in case of BGP takeover methods applying the virtualization, it is approximately 53% in the takeover. In regards to GR, due to RIB table update process occurring in the takeover, its CPU time consumption is about 60% in the takeover.

As to BGP takeover methods applying the virtualization, there is no need for the copying of backup data because of the following reasons: (1) BGP configuration information file (BGPd.conf) and (2) the data volume of the RIB table is shared. Therefore, even though the CPU time consumption in case of the takeover also increases, it decreases faster than that of BGP takeover methods not applying the virtualization as well as GR.

5.4 The results of simulation with functional improvements

This subsection describes the results of the simulation by improving the function of RM in the method to which the virtualization is not applied. In the case of the method not applying the virtualization, which was previously simulated, the backup information stored in the standby RM is copied in each layer after the failure detection, and configuration settings are updated in the process of the BGP takeover. During the copy and update process, the long latency occurs. If the information that standby RM has is stored in advance before the failure detection, the reduced latency can contribute to enhancing the performance.

Figure 14 shows the comparison in terms of throughput and latency among the method not applying the virtualization with or without modified RM, the method applying the virtualization with or without advance update under the condition of hold time = 0 s and RIB table entry = 100 thousand as simulated in the preceding section.

With respect to the throughput, the BGP takeover methods not applying the virtualization (with modified RM) showed around 19% difference compared to that with unmodified RM, and it is as good as that of the method applying the virtualization. However, proposed scheme can also be further improved by using the advance updating. Using the advanced updating, the method applying the virtualization, which is a modified virtualization, outperformed all other methods. The average throughput of the method not applying the virtualization (with modified RM) was about 779.81 Mbps and the method applying the virtualization (with modified virtualization) was about 854.98 Mbps, showing an 8.7% difference in performance.

The latency of the method not applying the virtualization with modified RM is also significantly reduced compared to the methods with unmodified RM, showing a difference of 0.1242 s (88.7% improvement). However, the method applying the modified virtualization showed a difference of 0.0098

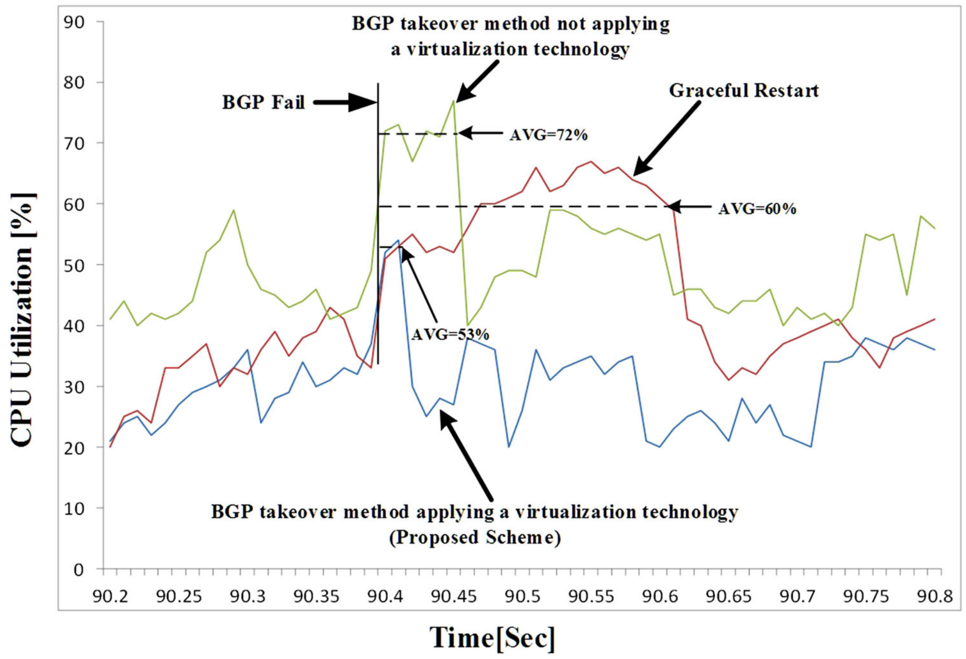


Fig. 13 Comparison of CPU time consumption

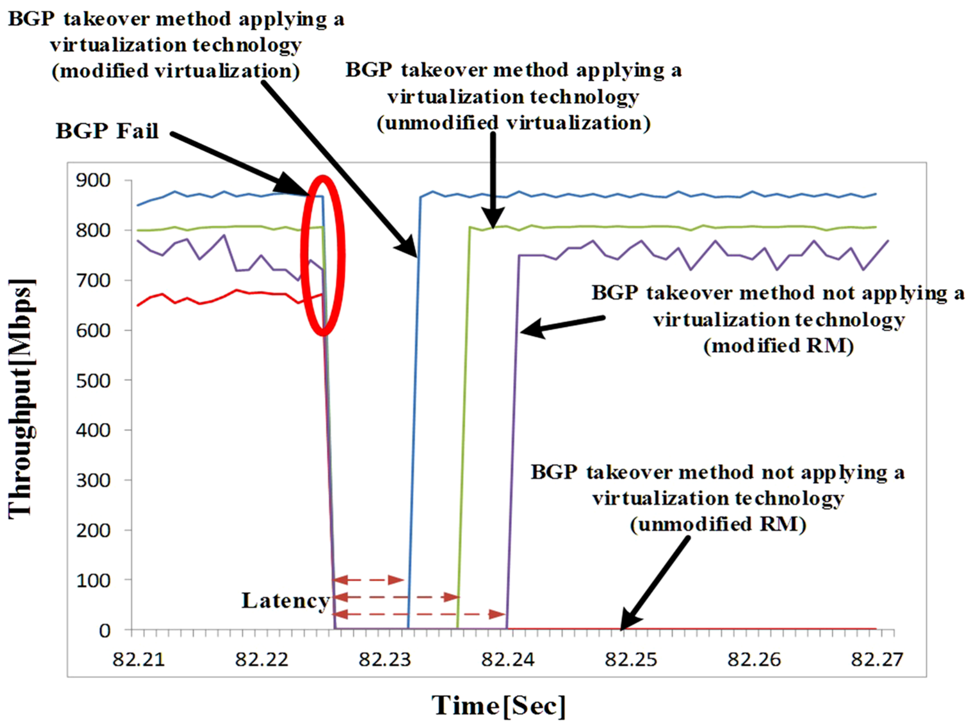


Fig. 14 Comparison of BGP throughput and latency (RIB table entry = 100 thousand, hold time = 0 s)

s compared to method not applying the virtualization with modified RM (62% improvement). The simulation results of BGP throughput & latency in each case is summarized in Table 4.

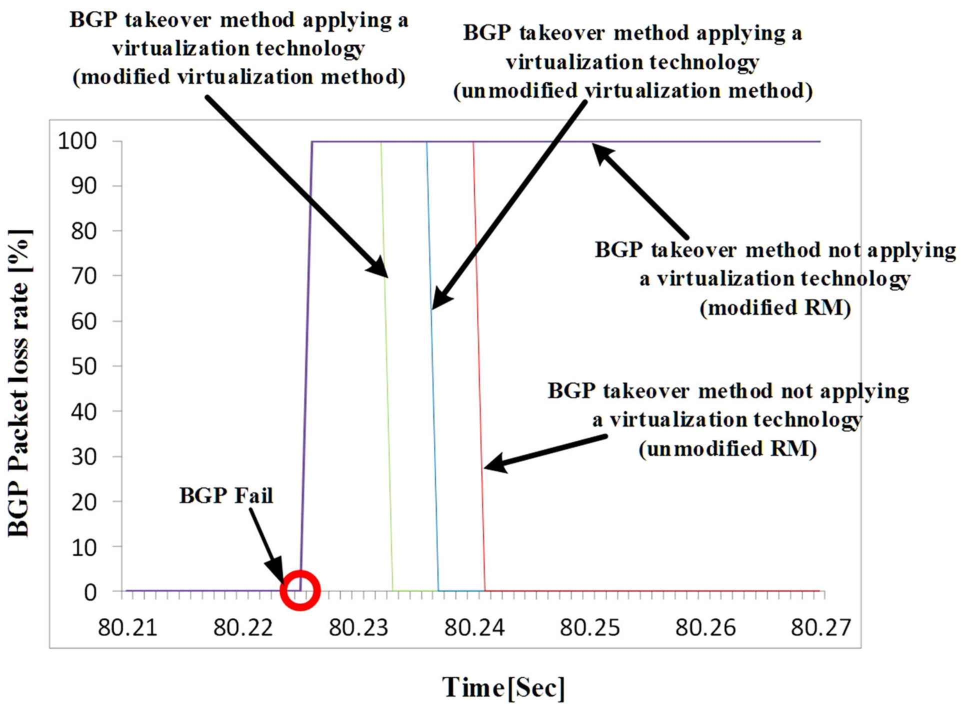
In the case of the packet loss, as shown in Fig. 15, the simulation revealed that as the latency of the method not applying the virtualization with modified RM and method applying the modified virtualization was reduced, the packet loss also decreased.

The method applying the virtualization proposed in this paper and GR operate in a single physical server. The fast BGP takeover is possible if the BGP takeover or recovery is occurred because of BGP problem. However, both methods require the reconnection of BGP and TCP in the case of the physical server shutdown. In case of the method not applying the virtualization, however, while BGP takeover time increases due to the data copy between the active and standby existing in the different physical servers, the

Table 4 Comparison of BGP average throughput and latency

	BGP takeover method applying a virtualization technology		BGP takeover method not applying a virtualization technology	
	Modified visualization method	Unmodified virtualization method	Modified RM	Unmodified RM
Throughput	854.98 Mbps (100%)	795.32 Mps (93%)	779.81 Mbps (91.2%)	622.56 Mbps{72.8%)
Latency	0.006 s	0.01 s	0.0158 s	0.14 s

Fig. 15 Comparison of BGP packet loss by time



problems occurred in the physical servers can be flexibly addressed.

Although not described in detail in this paper, in case of the proposed method applying the virtualization, the migration of active BGP currently operating to standby physical server is possible through the migration function of Docker. Also, the shutdown problem in the physical server can be flexibly handled by maintaining network sessions through the methods such as TCPR or IP tunneling. The details of migration will be described in another paper.

Also, the general performance of GR and the method applying the modified virtualization proposed in this study are similar. Yet, GR has a problem of not being able to set the new path during the recovery in case of the BGP failure since it goes through the normal BGP reconnection process in the control plane. However, the method applying the modified virtualization proposed in this paper is able to set the path faster than GR since it provides the faster BGP takeover.

6 Conclusion

This study investigated a BGP takeover method that supports high availability based on virtualization technique. First, we analyzed messages used in BGP, derived data to be backed up and points to store them, designed method for BGP takeover applying the virtualization technology, and then tested it by simulation.

As for data used to backup BGP session information data included in the BGP Open message, data included in the BGP Update message, state information, and routing tables were defined. In addition, the BGP takeover method based on the Docker was proposed. Since the proposed method creates two BGP daemons (Active, Standby) on one physical server, it has an advantage over the existing methods using two physical nodes in terms of cost savings.

The simulation also verified that through data volume and network stack sharing, it achieved a higher level of performance compared to that of the existing BGP takeover methods.

In addition, the results of simulation performed by improving the virtualized router confirmed that the performance of the method applying the virtualization technology can be further improved in terms of the throughput, latency, packet loss and cpu utilization.

For future work, we plan to conduct a research on the methods to support network high availability not only in the BGP protocol, but also in the virtualization infrastructure.

Acknowledgements This research was supported by the IT R&D program of MSIP (Ministry of Science, ICT and Future Planning) /IITP (Institute for Information & Communications Technology Promotion). [12221-14-1001, Next Generation Network Computing Platform Testbed] and The work reported in this paper was conducted during the sabbatical year of Kwangwoon University in 2015.

References

1. Alvisi, L., Bressoud, T., El-Khashab, A., Marzullo, K., Zagorodnov, D.: Wrapping server-side TCP to mask connection failures. In: Proceedings of IEEE INFOCOM, Vol. 2001 (2001)
2. Surton, R., Birman, K., van Renesse, R.: Application-driven TCP recovery and non-stop BGP. In: DSN '13 Proceeding of the 2013 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (2013)
3. The Linux-HA User's Guide. http://www.linux-ha.org/doc/users-guide/users-guide.html
4. Kim, H.S., Sang I.K.: A BGP session takeover method for high availability. In: ICUFN 2015 Seventh International Conference on Ubiquitous and Future Networks (2015)
5. RFC Recommendation 4271, A Border Gateway Protocol 4 (BGP-4) (2006)
6. Docker. https://docs.docker.com/userguide/dockerrepos/
7. exbgp. https://github.com/Exa-Networks/exabgp
8. Keller, E., Rexford, J., van der Merwe, J.: Seamless BGP migration with router grafting. In: Proceedings of the 7th USENIX Symposium on Networked Systems Design and Implementation (NSDI'10), San Jose, CA (2010)
9. Sahoo, A., Kant, K., Mohapatra, P.: Characterization of BGP recovery time under large-scale failures. In: IEEE International Conference on Communications (ICC'06), June 2006, pp. 949–954 (2006)
10. ns-BGP. http://www2.ensc.sfu.ca/~ljilja/cnl/projects/BGP-ns-2.34/ns-2.34ZBGP.html#Using_ns_BGP
11. Sreekumari, P., Jung, J.I., Lee, M.: A simple and efficient approach for reducing TCP timeouts due to lack of duplicate acknowledgments in data center networks. Clust. Comput. **19**(2), 633–645 (2016)
12. Choi, J., Ahn, Y., Kim, S., Kim, Y., Choi, J.: VM auto-scaling methods for high throughput computing on hybrid infrastructure. Clust. Comput. **18**(3), 1063–1073 (2015)
13. Xia, L., Cui, Z., Lange, J., Tang, Y., Dinda, P., Bridges, P.: Fast VMM-based overlay networking for bridging the cloud and high performance computing. Clust. Comput. **17**(1), 39–59 (2014)
14. Lo Cigno, R., Procissi, G., Gerla, M.: Sender-side TCP modifications: performance analysis and design guidelines. Clust. Comput. **8**(1), 35–45 (2005)

Hwasung Kim received the B.S. & M.S. degree in Electronic Engineering from Korea University, Seoul, Korea, in 1981 & 1983, and his Ph.D. degree in Computer science, from Lehigh University, Bethlehem, Pennsylvania, USA, in 1996. He worked as a senior researcher in Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea from 1984 to 2000. Currently, he is a professor in Kwangwoon University, Seoul, Korea since 2000. His research interests are Network Function Virtualization, Software Defined Networking, Mobile network protocol, Cloud Computing and High Performance Computing.

Sangil Kim received the M.S. in Electronic and Communication Engineering from Kwangwoon University, Seoul, Korea, in 2012. Currently, He is a Ph.D. candidate in Kwangwoon University, since 2012. His research interests are Network Function Virtualization, Software Defined Networking.

Hoyong Ryu received the M.S. and Ph.D. in Electronic and Communication Engineering from Kwangwoon University, Seoul, Korea, in 1995 and 1999, respectively. He has worked as a research team leader in Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea since 1998. His research interests are Network Operating System, Software Defined Networking, Network Function Virtualization and Computing Virtualization.

JaeHyung Park received his B.S. in computer science from Yonsei University, Korea, in 1991, and his M.S. and Ph.D. in computer science from Korea Advanced Institute of Science and Technology (KAIST), Korea, in 1993 and 1997, respectively. From 1997 to 1998, he was with the Center for Artificial Intelligence Research (CAIR) in KAIST. From 1998 to 2002, he was with the Network Laboratory in ETRI. Since 2002, he has been with the faculty of Chonnam National University, Korea, where he is currently an associate professor with the School of Electronics and Computer Engineering. His research interests are Internet Routing, Multicast Routing, Network Security, Wireless Mesh Networks, and Wireless Mesh/Ad-hoc Networks.

Jinsul Kim received the B.S. Degree in computer science from University of Utah, Salt Lake City, Utah, USA, in 2001, and the M.S. and Ph.D. degrees in digital media engineering, department of information and communications from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2005 and 2008. He worked as a researcher in IPTV Infrastructure Technology Research Laboratory, Broadcasting/Telecommunications Convergence Research Division, Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea from 2005 to 2008. He worked as a professor in Korea Nazarene University, Chon-an, Korea from 2009 to 2011. Currently, he is a professor in Chonnam National University, Gwangju, Korea. He has been invited reviewer for IEEE Trans. Multimedia since 2008. He was General Chair of IWICT2013/2014/2015, ICITCS2014, ICISA2015, ICMWT2015 and ICISS2015. His research interests include QoS/QoE, Measurement/ Management, IPTV, Mobile IPTV, Smart TV, Multimedia Communication and Smart Space/Works.