Improving Reliability for Multi-Home Inbound Traffic: MHLB/I Packet-Level Inter-Domain Load-Balancing

Hiroshi Fujinoki
Department of Computer Science
Southern Illinois University Edwardsville
Edwardsville, Illinois 62026-1656 USA
E-mail: *hfujino@siue.edu*

Abstract

Multi-homing is a network configuration that connects a customer network to multiple service providers. It is used to improve fault-tolerance and throughput. One of its problems is the lack of dynamic load-balancing for inbound network traffic to multi-homed networks, which prohibits us from taking advantage of multi-homing to improve reliability for inbound network traffic. This paper proposes a new routing architecture and a protocol, BGP-MHLB/I (BGP-Multi-Home Load Balancing/Inbound), to realize dynamic load-balancing for inbound traffic to multi-homed networks. Our analysis found approximately 80 multiple BGP paths available between two customer networks for up to two extra AS-hop paths. This finding suggests that the proposed BGP-MHLB/I routing will be an effective solution for improving reliability in the Internet.

1. Introduction

The lack of dynamic load balancing for inbound traffic in today's multi-homing stems from the nature of the path-vector routing and the lack of QoS support in Border Gateway Protocol version 4 (BGP4) . In this paper, the term "inbound network traffic" means network traffic as part of connections initiated by remote hosts. In BGP4, each autonomous system (AS) announces the range of IP addresses of the host computers that belong to the AS in "prefix" format, with its unique AS number using the message called UPDATE. Two neighboring ASes exchange UPDATE messages and chain-reactions of UPDATE message exchanges let each AS in the Internet learn memberships of individual hosts in other ASes and the paths to reach them [1]. However, each AS propagates only the best path selected by an AS farther to other ASes. This leads to two different types of losses in routing information: losses of inter-domain multiple BGP paths and losses of multiple access links to multi-homed destinations. Figure 1 demonstrates the two different types of losses in routing information.

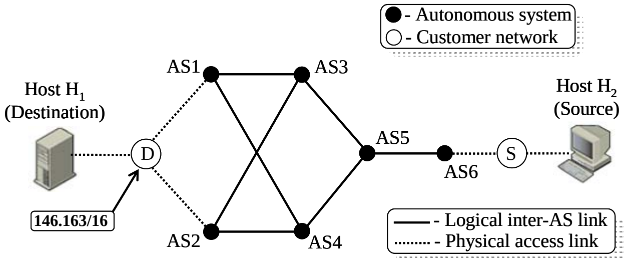


Figure 1. Two different types of losses in routing information caused by path-vector routing

In Figure 1, H_2 is a host computer connected to network *S*, which subscribes to a provider, AS6. Host H_1 is a host computer connected to another network *D*, who is multi-homed to two different providers, AS1 and AS2. Assume that *D* announces its prefix, 146.163/16 to both AS1 and AS2. The BGP speakers in AS1 and AS2 broadcast UPDATE messages to advertise *D*'s prefix to other ASes. AS5 receives the UPDATE messages through four different paths: AS1→AS3, AS1→AS4, AS2→AS3 and AS2→AS4. Since AS5 will select the best path to reach *D* (e.g., AS1→AS3) and it forwards only the best path to its downstream, AS6 can see only the best path selected by AS5 (e.g., AS1→AS3→AS5). When AS6 receives routing information to reach *D*, BGP loses: the routing information for multiple inter-domain paths and the availability of multiple access links to *D*.

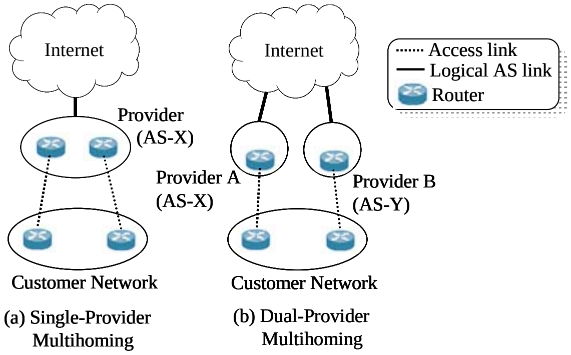
As a result, the two different types of losses in routing information by BGP, in conjunction with its lack of QoS support, make dynamic load-balancing infeasible for inbound network traffic to any multi-homed network. The above discussion also implies that inbound load balancing for multi-homed destinations requires load balancing by inter-domain routing.

The rest of this paper is organized as follows. Section 2 describes different implementations of multi-homing. Section 3 discusses the existing related work. Section 4 describes a new architecture and routing protocol, MHLB/I protocol, that realize inbound load balancing for multi-homed networks. Section 5 presents performance analysis. Section 6 summarizes the conclusions and on-going work, followed by a list of the selected references.

2. Multi-homing variants

Despite the simplicity in its concept, implementations of multi-homing are complex. Multi-homing can be categorized from the number of the providers a customer network is multi-homed to, how multi-homed networks are assigned network addresses, and types of functions performed in multi-homing.

A customer network can be multi-homed to a provider (called “single-provider multi-homing” shown by Figure 2-(a)) or to different providers (called “multi-provider multi-homing” shown by 2-(b)). Especially from the points of fault-tolerance and load balancing, the multi-provider multi-homing is a more attractive option. This paper assumes the multi-provider multi-homing.



(a) Single-Provider Multihoming

(b) Dual-Provider Multihoming

Figure 2. Single-provider and dual-provider multi-homing

Two major functions to improve fault-tolerance and throughput are load balancing and sharing. Load balancing is a technique that distributes both outbound and inbound network traffic to multiple access links to providers. Dynamic load balancing adjusts allocation of network traffic to each access link based on its current status, such as residual transmission bandwidth, end-to-end delay and packet-loss rate, while static load balancing allocates traffic load based on fixed pre-determined manual configurations. Load sharing is a technique that utilizes access links either for outbound or inbound traffic.

There are trade-off problems in the granularity of load balancing. Guo mentioned four possible levels of load balancing: packet-level, connection-level, host-level, and AS (or prefix)-level load balancing [2]. Table 1 summarizes the four levels of load balancing. In the packet-level load balancing, load balancing is performed for every single packet. Its primary advantages are maximum fault-tolerance against link failures and transmission bandwidth utilization of each access link. No state information is required except the current status for each access link. The primary problems are out-of-order packet deliveries, which can cause unnecessary packet-loss errors for connection-oriented transmissions and high computation overhead for routing each packet. Connection-level balancing selects access links for each unique combination of <source address, source port, destination address, destination port>. It provides fine granularity without out-of-order packet delivery, but requires a large volume of state information. Host-level balancing selects access links for each destination host address and it is a middle-ground between the connection-level and AS-level load balancing. The AS-level load balancing selects access links based on the destination AS numbers or prefixes, which does not cause out-of-order packet delivery and requires less state information than connection-level or host-level load balancing. Its primary disadvantage is coarse granularity, which prevents the fault-tolerance and maximizing utilization of each access link.

Table 1. Summaries of the four levels of load balancing

Load balancing types	Unit of balancing	Advantages	Disadvantages
Packet-level	Each packet	• Finest granularity • Stateless • Fault-tolerant	• Out-of-order packet delivery • High routing overhead
Connection-level	Each connection	• No out-of-order packet delivery	• Large volume of state information • Not fault-tolerant
Host-level	Each unique pair of source and destination IP addresses	• No out-of-order packet delivery	• Moderate granularity • State information still required • Not fault-tolerant
AS (or prefix)-level	Each destination AS number or destination prefix	• No out-of-order packet delivery	• Coarse granularity (not maximize link utilization) • Not fault-tolerant

Regarding how a multi-homed network is assigned network (i.e., IP) addresses, the three major options are provider-independent (PI), provider-aggregated (PA) and private addresses [3]. In the PI address, each customer network obtains a global network address space (as prefix) that does not belong to any one of its providers. With global PI addresses, both static and dynamic load balancing are theoretically possible in packet level for outbound traffic, but static load balancing is possible for inbound traffic only at the prefix-level (more detail is described in Section 3).

In PA address, each multi-homed network is assigned address space owned by one of its providers. Network address translation (NAT) or punching hole is used for transmitting outbound traffic through the provider(s) other than the one that owns the address space. NAT at the gateway to the other provider transforms the PA addresses to the one that belongs to the address space of the other provider [2]. With NAT, dynamic load balancing is possible for outbound traffic at or above the connection level. However, packet-level load balancing is not possible because the source address in the IP packets for a connection will be different for different access links. Inbound connections must go through the provider from whom a customer network obtains its PA addresses. Thus, load balancing for inbound traffic is not possible.

Punching hole is a technique that advertises the PA addresses allocated by one of the providers (let us call the provider who owns the PA addresses "the primary provider") also advertised through the other multi-homed provider(s). This approach allows remote hosts to initiate connections to local hosts in a multi-homed network through each provider. Regarding outbound traffic, since the source address in IP packets will not be modified no matter through which provider outbound packets are transmitted, static and dynamic load balancing are possible in all the four levels of load balancing. However, for inbound traffic, load balancing is not possible for the following reason. Since the punching-holed prefix is more specific than the aggregated prefix advertised through the primary provider, the longest match policy in BGP lets remote hosts to select the paths announced by punching hole. Thus, the prefix advertised through the primary provider will never be selected by remote hosts to reach the punching-holed addresses. CISCO introduced *BGP multipath* to solve this problem and it is discussed in Section 3. Another problem is that punching hole will result in a large number of specific prefixes, which will inflate BGP routing table.

Table 2. Summaries of multi-homing variants

Type of network addresses	Type of global addresses	Announcing methods	Potential outbound load balancing	Potential inbound load balancing
Global address	PI address	• PI-AS number	• Static and dynamic • Packet-level	• Static only • Prefix-level
		• PA-AS number	• Static and dynamic • Packet-level	• Static only • Prefix-level
	PA address	• NAT	• Static and dynamic • Connection-level	N. A.
		• Punching hole	• Static and dynamic • Connection-level	• Static only • Prefix-level
Private address	N. A.	• NAT	• Static and dynamic • Connection-level	N. A.

In private address configurations, each multi-homed network is assigned private addresses. Each interface in a gateway that is directly connected to a provider is assigned a PA address of the provider and typically NAT transforms a private address of a local host to a PA address of a provider for outgoing packets. Static and dynamic load balancing are possible at or above the connection-level. Load balancing for inbound traffic is not possible by the same reason as NAT/PA-address combination. Table 2 summarizes the various implementations of multi-homing described above regarding the potential for outbound and inbound load balancing. This paper proposes a new architecture and a routing protocol that realize packet-level dynamic load balancing for inbound traffic to multi-homed customer networks that use PI addresses or PA addresses announced by punching hole (shown by shaded options in Table 2).

3. Related existing work

For load balancing inbound traffic, prefix-level static load balancing has been used. In prefix-level static load balancing, the address space is partitioned to multiple sub-networks. Then, the prefixes for the sub-networks are separately announced to the Internet through different access links. This allows remote hosts to reach destination hosts in the different sub-networks through different access links [4]. This prefix-level load balancing is applicable to customer networks with PI addresses and PA addresses with punching hole (the shaded options in Table 2). There are two major problems. First, this approach offers only prefix-level static load balancing for inbound traffic. The second problem is that partitioning a customer network to smaller sub-networks causes inflation of BGP routing table size.

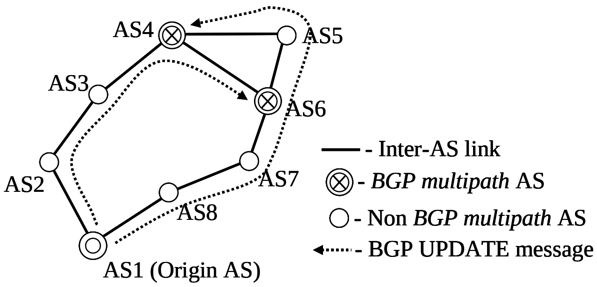


Figure 3. Possible routing loops due to multipath transmissions

CISCO introduced *BGP multipath* that allows BGP speakers to select multiple paths from local *Adj-RIB-in* table to reach destination ASes [5]. Since *BGP multipath* does not advertise multiple paths to other ASes, routing loops can happen if packet-level load balancing is performed. Figure 3 shows an example. AS1 advertises its prefix by UPDATE messages to AS2 and AS8. AS2 forwards the UPDATE message through AS3, AS4 and AS6. AS8 forwards the UPDATE message through AS7, AS6, AS5 and AS4. AS4 and AS6 implement *BGP multipath*. AS6 gets two paths to AS1: through AS4 and AS7. Similarly, AS4 gets two paths to AS1: through AS3 and AS5. When AS6 sends IP packets to AS1, it can transmit packets through AS4 and AS7. Since AS4 also has two paths to AS1 (through AS3 and AS5), the packets AS6 transmits to AS1 can be forwarded to AS5 by AS4, and AS5 forwards the packets to AS6. This causes a routing loop. This example implies that the more BGP routers implement *BGP multiplath*, the more likely routing loops can happen. Similarly, since multipath BGP speakers do not communicate with each other, network traffic can be split to a large number of paths through a couple multipath BGP speakers. The third problem is that since *BGP multipath* does not consider current traffic load or delay for each path, efficient dynamic load balancing will be difficult.

4. MHLB/I routing for inbound dynamic load balancing

This section describes a routing architecture and a protocol, BGP-MHLB/I (BGP-Multi-Home Load Balancing /Inbound), which extends MBGP protocol for multiple BGP path routing [6]. BGP-MHLB/I protocol performs the following functions:

- Dynamic load balancing that utilizes multi-homed access links for inbound network traffic
- Dynamic load balancing for BGP inter-domain routing that benefits multi-home access links
- Packet-level load balancing that maximizes reliability and bandwidth utilization of each access link

BGP-MHLB/I (called "MHLB/I" hereafter) routing consists of Multi-Home Management Agents (MHMA), Multi-Home Management Base (MHMB), Multi-Home Management Device (MHMD), and MHLB/I routers.

Multi-Home Management Agent (**MHMA**): A process that runs at the gateway routers in the source (transmitting) and destination customer networks. Each MHMA monitors network traffic, detects, receives, and transmits MHLB/I messages at a gateway router on behalf of the MHMB to its interfacing provider.

Multi-Home Management Base (**MHMB**): MHMB is a process that is responsible for implementing MHLB/I protocol at a gateway router in an end customer network. MHMB is running in Multi-Home Management Device (MHMD), which is a hardware device that is connected to local gateway routers. MHMB receives/transmits MHLB/I messages through local MHMA's and coordinates MHLB/I routing with the MHMB in remote customer networks. MHMB is responsible for initiating MHLB/I routing and detecting/maintaining multiple BGP paths by exchanging MHLB/I messages with a remote MHMB. MHLB/I messages are all transferred using UDP datagrams through a particular port used by MHLB/I protocol.

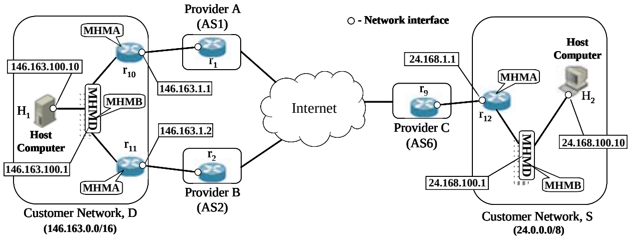


Figure 4. Organization of MHLB/I architecture

Figure 4 shows the organization of MHLB/I routing. A customer network, *D*, is multi-homed to two providers. Both source and destination customer networks require the same MHLB/I organization. In customer network, *D*, one of its hosts is assigned 146.163.100.10. Its MHMD is assigned 146.163.100.1. The interface that is directly connected to provider *A* is assigned 146.163.1.1 and the one for provider *B* is assigned 146.163.1.2. Similarly, the addresses in *S* are assigned as shown in the figure.

MHLB/I routers: BGP speakers that implement MHLB/I protocol. MHLB/I routers discover multiple BGP paths and access links, dynamically balance network traffic from a source to a destination customer network and manage the multiple BGP paths on demand of the source MHMB. MHLB/I protocol does not require every BGP speaker in the Internet to be a MHLB/I router.

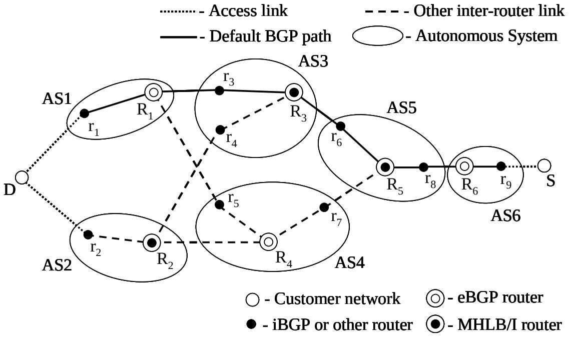


Figure 5. Simplified visualization of MHLB/I architecture

Figure 5 shows an example of MHLB/I routers deployed in the Internet. A customer network, *D*, is multi-homed to two providers, AS1 and AS2, through router r_1 and r_2 respectively, while *S* is single-homed to AS6 through r_9. Routers R_2, R_3, and R_5 are MHLB/I routers while R_1, R_4 and R_6 are eBGP (exterior BGP) routers that do not speak MHLB/I. The path shown by the solid links indicates the default BGP path from AS6 to *D*. Other links in dotted lines indicate other inter-router links. The default BGP path is the one the existing BGP selected to reach a destination network. Routers that are not an eBGP or MHLB/I router are shown as r_x (r_1 through r_9).

MHLB/I routing is implemented by four phases of MHLB/I initiation, multi BGP-path discovery, MHLB/I transmissions and termination. Each of the four phases are described in the following sections.

Phase 1: Initiation

MHLB/I routing is initiated by the MHMD in a source customer network. The source MHMD transmits INITIATE_MHLB message to the network (IP) address of a host computer in the destination network with the IP address of the source MHMD ($IP_{MHMD\text{-}S}$) and that of the destination host computer (IP_{DH}) in the source and destination address fields of the UDP packet header. Each INITIATE_MHLB message is in the payload field of a UDP packet and consists of the following elements:

- M_{LABEL}: "INITIATE_MHLB" character string
- Q_{NUM}: A unique message sequence number
- $IP_{SH\text{-}PREFIX}$: The source customer network address prefix
- $IP_{DH\text{-}PREFIX}$: The destination customer network address prefix

After the source MHMB initializes each field in an INITIATE_MHLB message, it transmits the message to a destination host through the default BGP path. Each router in the default BGP path forwards this message as a UDP packet. The MHMA at a gateway in the destination network captures every UDP packet that carries the destination port number used by MHLB/I and forwards them to its MHMD. When the message reaches the MHMD, the MHMB intercepts the message (the MHMB does not forward the message any farther). The MHMB creates a MHMB_ECHO message and sends it to $IP_{MHMD\text{-}S}$. Each MHMB_ECHO consists of the following elements.

- M_{LABEL}: "MHMB_ECHO" character string
- Q_{NUM}: The value used in the INITIATE_MHLB message
- $IP_{SH\text{-}PREFIX}$: The value of $IP_{SH\text{-}PREFIX}$ in INITIATE_MHLB
- $IP_{DH\text{-}PREFIX}$: The value of $IP_{DH\text{-}PREFIX}$ in INITIATE_MHLB
- $IP_{MHMD\text{-}D}$: The IP address of this MHMD
- $L_{GATEWAY}$: A list of IP addresses of the gateway routers in the destination network and the provider's AS number to which the gateway router is directly connected to

For the example shown by Figure 4 and 5 where H_2 transmits to H_1, $IP_{MHMD\text{-}S}$ and IP_{DH} in INITIATE_MHLB are assigned 24.168.100.1 and 146.163.100.10 by the MHMB in *S*. IP_{SH_PREFIX} and $IP_{DH\_\ PREFIX}$ are assigned 146.163.0.0/16 and 24.0.0.0/8. The message is propagated through the default BGP path towards H_1 in *D*. When the message reaches the MHMD in *D*, its MHMB intercepts it and responds with a MHMB_ECHO whose $IP_{MHMD\text{-}D}$ is 146.163.100.1 and $L_{GATEWAY}$ contains {(146.163.1.1, AS1), (146.163.1.2, AS2)}. The source and destination address fields in the UDP packet for the MHMB_ECHO message are $IP_{MHMD\text{-}D}$ and $IP_{MHMD\text{-}S}$.

Phase 2: Multi BGP path discovery

When the MHMB_ECHO comes back to the source MHMD, the first phase is completed and the source MHMB creates a DISCOVER_PATH message that has the following fields. The source MHMB adds its identification (AS number of its provider and IP address of its MHMD) and that of the next-hop router with the current time stamp at the beginning of L_{ASL}. Then the source MHMB broadcasts the message to $IP_{MHMD\text{-}D}$.

- M_{LABEL}: "DISCOVER_PATH" character string
- Q_{NUM}: The value used in the INITIATE_MHLB message
- $L_{DEST\_INTERFACES}$: The copy of $L_{GATEWAY}$ in the MHMB_ ECHO message
- L_{ASL}: List of inter-AS links that forms a BGP path
- $IP_{SH\text{-}PREFIX}$: The value of $IP_{SH\text{-}PREFIX}$ in INITIATE_MHLB
- $IP_{DH\text{-}PREFIX}$: The value of $IP_{DH\text{-}PREFIX}$ in INITIATE_MHLB
- C_{SPRIT}: Counter of BGP path splits (described later)

Every MHLB/I router watches the UDP port used by MHLB/I protocol (its port number is represented by *N*) and executes the procedure shown in Figure 6. In the figure, *U* represents a UDP packet and *K* represents a DISCOVER_PATH message carried by *U*. ID_X indicates identification of router *X* as a pair of its IP address and AS number. Similarly, IP_{THIS} represents the IP address of a MHLB/I router. *U.PORT* indicates a UDP port number. IP_{DEST_INT} is an address of a destination interface in $L_{DEST\_INTERFACES}$. The "+" operator indicates concatenation at the end of L_{ASL}, while "←" indicates assignment.

In the example shown by Figure 4 and 5, the MHMB in *S* constructs a DISCOVER_PATH message with L_{ASL} = {(r_{12}, r_9, t_0)}, where each router identification consists of its AS number and IP address. For example, "r_{12}" actually means (AS6, 24.168.1.1). "t_0" is the time stamp at the source MHMB. Then, the message is forwarded to r_9 through UDP port *N* at the source MHMD. If a source network is multi-homed, a DISCOVER_PATH message will be transmitted from each gateway router. The gateway router and its next-hop router are added at the beginning of L_{ASL} in a DISCOVER_PATH message. Since r_9 is not a MHLB/I router, it forwards the UDP packet to its next-hop router in the default BGP path, which is R_6. Since R_6 is not a MHLB/I router, it forwards the UDP packet to r_8, in the same way as r_9. The next router r_8 handles the UDP packet in the same way.

```
1. if ($U.PORT_{DEST} = N$) and ($U.PORT_{SOURCE} = N$) and ($U.K.M_{LABEL}$ = "PATH_DISCOVERY"), go to step 2.  Otherwise forward $U$ to the next router in the default BGP path to $U.IP_{DH}$ and terminate
2. Test if $IP_{THIS} \notin U.K.L_{ASL}$.  If the test holds true, proceed to step 3.  Otherwise terminate.
3. for (each outgoing link at this router)
4.     if (there is a BGP path through which any $IP_{DEST\_INT} \in U.K.L_{DEST\_INTERFACES}$ can be reached)
5.         Duplicate $K$ to $P$
6.         $P.L_{ASL} \leftarrow P.L_{ASL} + (ID_{THIS}, ID_{NEXT-ROUTER}, t_x)$ and copy $P$ to a new UDP packet, $U_{NEW}$
7.         Forward $U_{NEW}$ through the outgoing link
8.         Delete $P$ and $U_{NEW}$
9.     end-if
10. end-for
```

Figure 6: The procedure to process a DISCOVER _PATH message at a MHLB/I router

After r_8 forwards the DISCOVER_PATH message to R_5, R_5 will see the message in a UDP packet (step 1) and proceeds to step 2. Since this message goes through R_5 for the first time, R_5 should not appear in $K.L_{ASL}$ (thus the condition in step 2 is met) and it proceeds to step 3. Since R_5 has two AS links through which either 146.163.1.1 or 146.163.1.2 can be reached (through AS3 or AS4), it transmits the message to both r_6 and r_7. When R_5 forwards the DISCOVER_PATH through r_6, $L_{ASL} = \{(r_{12}, r_9, t_0), (R_5, r_6, t_1)\}$. Similarly, L_{ASL} through r_7 contains $\{(r_{12}, r_9, t_0), (R_5, r_7, t_1)\}$.

Since r_6 is another non-MHLB/I router, it forwards the DISCOVER_PATH message to R_3 without any processing. When R_3 sees the UDP packet, it performs the procedure in Figure 6. Since R_3 has two paths through which either 146.163.1.1 or 146.143.1.2 can be reached, it forwards the message to both r_3 and r_4. For the one to r_3, L_{ASL} contains $\{(r_{12}, r_9, t_0), (R_5, r_6, t_1), (R_3, r_3, t_2)\}$ and the one to r_4 contains $\{(r_{12}, r_9, t_0), (R_5, r_6, t_1), (R_3, r_4, t_2)\}$. Since R_4 is not a MHLB/I router, when it receives a DISCOVER_PATH message from r_7, it forwards the message only to the next router in its default BGP path to IP_{MHMD-D} (e.g. R_2). When R_2 receives the message, it forwards the message to both r_4 and r_2. When the MHMA in one of the gateway routers in D sees a DISCOVER_PATH message, it forwards the message to the MHMD. Assume that the destination MHMD receives the following four DISCOVER_PATH messages, each of which has a unique L_{ASL}. The messages (a) and (d) arrive at the destination MHMD through r_{10} and (b) and (c) through r_{11}.

(a) $(r_{12}, r_9, t_0), (R_5, r_6, t_1), (R_3, r_3, t_2)$
(b) $(r_{12}, r_9, t_0), (R_5, r_6, t_1), (R_3, r_4, t_2), (R_2, r_2, t_3)$
(c) $(r_{12}, r_9, t_0), (R_5, r_7, t_1), (R_2, r_2, t_4)$
(d) $(r_{12}, r_9, t_0), (R_5, r_7, t_1), (R_2, r_4, t_4), (R_3, r_3, t_5)$

When a MHLB/I router accepts a DISCOVER_PATH message at step 2 in Figure 6, it constructs the MHLB/I routing table (Table 3).

Table 3. MHLB/I routing table at each MHLB/I router

Source Address ($IP_{SH-PREFIX}$)	**Destination Address ($IP_{DH-PREFIX}$)**	**Next-hop Routers**
146.163.0.0/16	24.0.0.0/8	(blank)

Each time the destination MHMB receives a DISCOVER_ PATH message, it transmits L_{ASL} in the message to the source MHMB using PATH_NOTIFY message. Each PATH_NOTIFY message consists of:

- M_{LABEL}: "PATH_NOTIFY" character string
- Q_{NUM}: The value used in the INITIATE_MHLB message
- $IP_{DH-PREFIX}$: The value of $IP_{SH-PREFIX}$ in INITIATE_MHLB
- $IP_{SH-PREFIX}$: The value of $IP_{DH-PREFIX}$ in INITIATE_MHLB
- L_{ASL}: A copy of the L_{ASL} in a DISCOVER_PATH

The source MHMB receives one PATH_NOTIFY for each unique path detected by a DISCOVER_PATH message. When the source MHMB receives a PATH_NOTIFY, it announces a detected path to each involved MHLB/I router using PATH_SETUP messages. For example, the source MHMB will send three PATH_SETUP messages to R_3, each of which contains (R_3, r_3, t_2), (R_3, r_4, t_2) or (R_3, r_3, t_5). When R_3 receives three PATH_SETUP messages, R_3 completes "Next-hop Routers" in the MHLB/I routing table by setting "r_3, r_4". The source MHMB performs the same for every MHLB/I router in each PATH_NOTIFY message.

For each multiple BGP path, allocation of network traffic load will be computed based on the observed round trip delay. When MHLB/I routers receive multiple PATH_SETUP messages with the same unique tuple of $<Q_{NUM}, IP_{DH-PREFIX}, IP_{SH-PREFIX}>$ from a source MHMB, they calculate the difference between a timestamp in a PATH_SETUP message and the MHLB/I router's current time. For example, since R_3 receives three PATH_SETUP messages (one for each multiple BGP path), MHLB/I calculates $\Delta t_a = T_a – t_2$, $\Delta t_b = T_b – t_2$, and $\Delta t_d = T_d – t_5$ where T_a, T_b, and T_d represent the timestamp at R_3 on the arrival of the PATH_SETUP message for (a), (b) and (d) respectively. If two different paths have the same next-hop router from a MHLB/I router, whichever shorter delta time is used for routing. For example, path (a) and (d) have the same next-hop router from R_3. If $\Delta t_a < \Delta t_d$, Δt_a is used for transmissions through r_3.

The traffic load weight for a BGP path, *k*, specifies the percentage of the traffic load assigned to path *k* to the total traffic load from $IP_{SH\text{-}PREFIX}$ to $IP_{DH\text{-}PREFIX}$ at a MHLB/I router. The traffic load weight is calculated using the following four steps:

Step 1: In *m* multiple paths detected at a MHLB/I router, the path with the shortest round trip delay is called the primary path. If there is only one path, it is the primary path and 100% of the traffic load is assigned to the path. Then the procedure is terminated. Otherwise proceed to Step 2.

Step 2: For each of the (*m*-1) non-primary paths, the ratio of their round trip delay to that of the primary path (designated as λ_1 through $\lambda_{(m\text{-}1)}$) is calculated by dividing the round trip delay of a non-primary path, Δt_k, by that of the primary path, Δt_i, (thus $1.0 \leq \lambda_k$ for any non-primary path *k*: $1 \leq k \leq (m\text{-}1)$).

Step 3: An index to determine the traffic load weight, called the weight score, is calculated for each path. For the primary path, a weight score of 100 is always assigned. For other paths, the weight score is calculated as: if λ_k is at or less than the lower threshold, λ_{Low}, a weight score of 100 is assigned to path *k*. If it is higher than the upper threshold, λ_{High}, then a score of 0 is assigned. If λ_k is between λ_{Low} and λ_{High}, the weight score is assigned inversely proportional to the distance between λ_{Low} and λ_{High}, using formula (1). For example if λ_k is 1.5 and $\lambda_{Low} = 1.2$ and $\lambda_{High} = 2.0$, the score for path *k* will be 62.5.

$$100 \times (((\lambda_{High} - \lambda_{Low}) - (\lambda_k - \lambda_{Low})) / (\lambda_{High} - \lambda_{Low})) \quad (1)$$

Step 4: Add the weight scores for all the *m* multiple paths and the traffic load weight is determined as a ratio of a score for a path to the total score. For example, if there are four paths and the weight scores for the three non-primary paths are 30, 50 and 70, the total weight score will be 250 (30+50+70+100). The traffic load weights for the four paths will be 12% (30/250), 20%, 28% and 40% of the traffic load (packets) from $IP_{SH\text{-}PREFIX}$ to $IP_{DH\text{-}PREFIX}$ at this MHLB/I speaker. Selecting only the paths whose end-to-end delay is within a particular threshold, errors due to out-of-order packet deliveries can be minimized for connection-oriented protocols, such as TCP.

Some ASes have a couple hundreds of links to other ASes [7]. This implies that uncontrolled flooding of DISCOVER_PATH messages can cause serious message overhead. To limit the message overhead, each DISCOVER_PATH message contains a field called "counter of BGP path splits". This field (C_{SPRIT}) is initialized by a positive integer before a DISCOVER_PATH message is broadcasted by the source MHMB. Each MHLB/I router decreases the counter, if the DISCOVER_PATH message is sent to more than one next-hop router. For example, at R_3 in Figure 5, since the message is duplicated to two next-hop routers, the counter is decreased by 1 (= 2–1). It would have been reduced by 2, if a message were duplicated to three paths.

If a MHLB/I router has more outgoing BGP paths than C_{SPRIT}, some BGP paths will be selected and C_{SPRIT} is set to 0. Once the counter reaches 0, any MHLB/I router in the downstream will not duplicate the DISCOVER_PATH message any more. Although the split counter is used as an approximate control, it will avoid uncontrolled flooding.

Phase 3: MHLB/I transmissions

After the MHLB/I routing is started, the source MHMB periodically broadcasts REFRESH messages. The message consists of the same contents as DISCOVER_PATH except that "counter of BGP path splits" field is not included. The message is propagated in the same way as DISCOVER_PATH messages. When a REFRESH message reaches a MHLB/I router, the router finds the next-hop routers from the "Next-Hop Routers" field in its MHLB/I routing table and forwards the refresh message only to those routers. The destination MHMB processes each REFRESH message in the same way as a DISCOVER_PATH message. When the source MHMB receives a PATH_NOTIFY message for a REFRESH message, it updates each involved MHLB/I router by a PATH_SETUP message.

Using PATH_SETUP messages to refresh Δt_x for each path *x* at a MHLB/I router, MHLB/I routers dynamically update the traffic load weights based on the latest round trip delay. If a path fails, the source MHMB detects the failure by lack of PATH_NOTIFY and will notify the failure to every affected MHLB/I router. The MHLB/I routers then drop the failed path and forward the next REFRESH message to another next-hop router to the destination (if any), which dynamically activates a new BGP path.

REFRESH messages will be broadcasted with a certain interval (e.g., 1 second). It is the interval changes in the round trip delay and link failures are detected, which allows MHLB/I routers to adjust traffic load or detour a failed path.

Phase 4: Termination

Termination of a MHLB/I routing is performed by TERMINATE message. The source MHMB broadcasts TERMINATE messages in the same way as REFRESH messages. When a MHLB/I router sees a TERMINATE message, it deletes the corresponding entry from its MHLB/I routing table.

5. Performance analysis

The number of multiple BGP paths from ASes to the core of the Internet was analyzed. We defined the core of the Internet as the AS that had the largest degree of AS inter-connections, which was AS 701 [7]. Figure 7 shows the results of our analysis based on the *Adj-RIB-in* table owned by AS 65000 [8], but similar results were observed for other ASes. In Figure 7, the numbers of distinct BGP paths from each of 62 ASes to AS 701 are shown for multiple BGP paths with 0 extra AS hop (the minimum-hop BGP paths), 1 extra and 2 extra hops. We recognized each BGP path by a sequence of AS numbers in each path.

We classified ASes to core or non-core ASes. The core ASes were those that had at least two local links that lead to AS 701, while non-core ASes had only one such a local link. Our analysis detected 62 core ASes in the routing table of AS 65000. We found that 54.8% of the 62 core ASes (34 ASes) had multiple (two or more) shortest BGP paths to AS 701 (the average was 2.2 paths) while the largest number was 9 multiple paths. For multiple paths with up to 1 extra AS hop, the average was 14.3 paths and 64.5% of the 62 ASes had 12 or more paths to AS 701. For up to 2 extra hops, the average was 80.5 paths and 96.7% of the core ASes had 20 or more paths and every AS had at least 9 paths (the largest number was 356 paths). These results suggest that multiple BGP paths are available for most of the cases and multiple-path transmissions can be an effective solution to maximize network resource utilization and reliability.

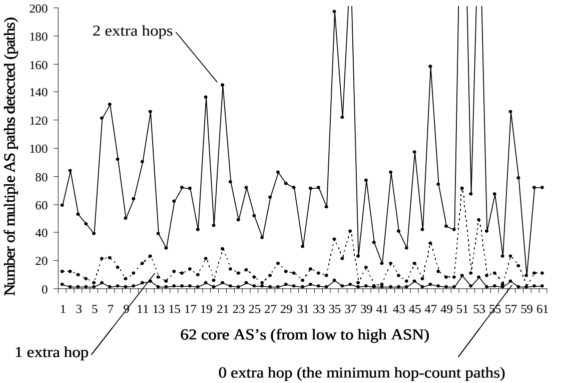


Figure 7. Number of multiple BGP paths detected at AS 65000

Regarding fault-tolerance against link failures, assume that there are *m* multiple paths from *S* to *D* through each access link of *D*. Let us assume that the probability of failure for each link in a particular time interval is *p* ($0 < p < 1.0$). We assumed that each BGP path consists of three sections: ① links from the source provider to the first MHLB/I router where multiple BGP path routing starts, ② links between the first MHLB/I router and the last MHLB/I router where multiple paths exist and ③ links between the last MHLB/I router where multiple path routing ends and a destination provider. Figure 8 shows this structure.

In the example, the destination network *D* is multi-homed to two providers, while *S* is single-homed to a provider. R_{15} is the router in the source provider that interfaces to *S*. Similarly, R_1 and R_7 are the routers in the destination providers that interface to *D*. The section between R_{15} and R_{13} corresponds to ① and the section is assumed to have χ inter-router links. Similarly, the sections between R_{13} and R_3 and between R_{13} and R_9 correspond to ②, each of which has β links. The sections between R_3 and R_1 and between R_9 and R_7 correspond to ③, each with α links.

For a BGP path, all the links must be working for successful transmissions of network traffic. Let *n* stand for the degree of multi-homing ($n = 2$ in Figure 8). Thus, the expected probability for successful transmission will be $(1\text{-}p)^{(\alpha+\beta+\chi)}$ for the existing BGP. For MHLB/I, since as long as one of the paths is available, transmission can be sustained, the probability for successful transmission using MHLB/I is predicted by formula (2).

$$(1\text{-}((1\text{-}(1\text{-}((1\text{-}((1\text{-}p)^{\beta}))^{m})) \times ((1\text{-}p)^{\alpha}))^{n})) \times ((1\text{-}p)^{\lambda}) \qquad (2)$$

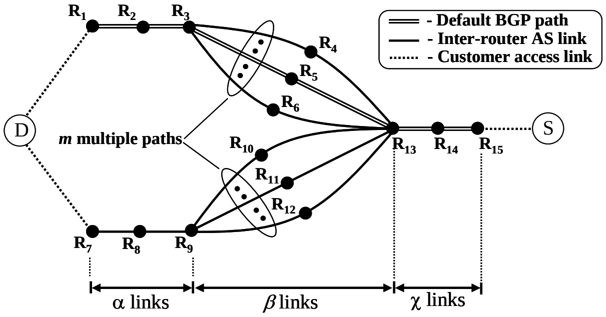


Figure 8. *m* multiple BGP paths for each of *n* multi-homed access links

Table 4 summarizes the major processing complexity for each MHLB/I message. MHMB_ECHO requires *O*(*N*) at the destination MHMB where *N* represents the degree of multi-homing at a destination network. For DISCOVER_PATH, *O*(*N*) is required at the source MHMB while *N* represents the degree of multi-homing at a source network since the message needs to be transmitted from each multi-homed access link. *O*(*MN*) is required for each DISCOVER_PATH message at a MHLB/I router where M is the number of outgoing links at the MHLB/I router to N distinct ISP's the destination network is multi-homed to. Each PATH_NOTIFY requires $O(N)$ at the source MHMB where N represents the number of MHLB/I routers in L_{ASL} in a PATH_NOTIFY. REFRESH and TERMINATE require the same cost as DISCOVER_PATH. Other activities cost constant time.

Table 4. Summaries for algorithm complexity to process MHLB/I messages

Message Types	Source MHMB	MHLB/I Routers	Destination MHMB
INITIATE_MHLB	$O(1)$	N.A.	$O(1)$
MHMB_ECHO	$O(1)$	N.A.	$O(N)$
PATH_DISCOVERY	$O(N)$	$O(MN)$	$O(1)$
PATH_NOTIFY	$O(1)$	N.A.	$O(N)$
PATH_SETUP	$O(1)$	$O(1)$	N.A.
REFRESH	$O(N)$	$O(MN)$	$O(1)$
TERMINATE	$O(N)$	$O(MN)$	$O(1)$

The delay before load balancing by MHLB/I starts, including those for discovering destination MHMB, discovering multiple paths and activating them for load balancing, is transparent from the transmitting hosts. MHLB/I protocol discovers multiple paths and sets them up while payload traffic is being transmitted through the default BGP path and as multiple paths are being dynamically detected, the payload traffic is transparently diverted to the detected multiple paths.

6. Conclusions and future work

This paper proposes a new architecture and a routing protocol that realize packet-level load balancing for inbound traffic to maximize the benefits from multi-homed networks. The complexity for processing MHLB/I messages at a MHLB/I router will be $O(mn)$. MHLB/I routing protocol is designed so that it hides delay before multiple BGP paths are detected and multi-home load balancing starts.

The new protocol achieves the goals by extending the existing BGP and it requires only limited deployment of the MHLB/I routers in the Internet. Our analysis on the degree of multiple paths between any two ASes in the Internet suggests that multiple-path transmissions can be an effective solution to maximize network resource utilization and reliability. Ge found that more tier-2 providers are horizontally interconnected to each other [9]. All these findings suggest that the proposed MHLB/I protocol will be an effective solution to take the advantage of the recent growth in the Internet. Currently, we are developing simulations that measure the improvement in reliability, bandwidth utilization and the messaging overhead using the real Internet structure based on *Adj-RIB-in* tables published by several providers [10].

References

[1] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)," *RFC 1771*March 1995.

[2] F. Guo, J. Chen, W. Li, and T. Chiueh, "Experiences in Building A Multihoming Load Balancing System," *Proceedings of IEEE INFOCOM*, 2004, pp. 1241-1251.

[3] R. Gummadi and R. Govindan, "Practical Routing-Layer Support for Scalable Multihoming", *Proceedings of IEEE INFOCOM*, 2005, pp. 248-259.

[4] A. Akella, A. Shaikh, and R. Sitaraman, "A Measurement-Based Analysis of Multihoming," *Proceedings of ACM SIGCOMM*, 2003, pp. 353-364.

[5] M. Tufail, "IPv6 - An Opportunity for New Service and Network Features," *Proceedings of the International Conference on Networking and Services*, 2006, pp. 11.

[6] H. Fujinoki, "Multi-Path BGP (MBGP): A Solution for Improving Network Bandwidth Utilization and Defense against Link Failures in Inter-Domain Routing," to appear in *Proceedings of IEEE International Conference on Networks 2008*.

[7] *CIDR Report*, May 2008: http://www.cidr-report.org/

[8] BGP Table Statistics, http://bgp.potaroo.net/as2.0/bgp-active.html

[9] Z. Ge, D. Figueiredo, S. Jaiswal, and L. Gao, "On the Hierarchical Structure of the Logical Internet Graph," *Proceedings of SPIE ITCom*, vol. 4526, 2001, pp. 208-222.

[10] University of Oregon Route Views Project, July 2008: http://www.routeviews.org/