# PLUG AND PLAY SERVER LOAD BALANCING AND GLOBAL SERVER LOAD BALANCING FOR TACTICAL NETWORKS

**William V. Wollman, Harry Jegers,**
**Maureen Loftus, Caleb Wan**

**The MITRE Corporation**
**12 Christopher Way**
**Eatontown, NJ 07724**

## ABSTRACT

*Server Load Balancing (SLB) technology as described by [1] has provided a fundamental cornerstone necessary to the success of the Internet. Global Server Load Balancing (GSLB) as described in [2, 3] enhances application performance by allowing the client's server selection to be based upon server performance, network performance and the approximate network location of the client and the server. To help enable this technology for the tactical environment we have created the Server Load Balancing Registration Protocol* [1].

## INTRODUCTION

This Server Load Balancing Registration Protocol allows servers to register and coordinate their services with a Server Load Balancer (SLB) manager application. The SLB manager uses these registration messages to automatically configure a SLB to support the services provided by a real server. Coordination between real servers and the SLB manager will include service definition(s), dynamically assigned or pre-planned static Virtual IP (VIP) addresses, and server application health monitoring. The SLB manager will also register dynamically-assigned VIP addresses with the Domain Name System (DNS)-based GSLB, support VIP address route injection, and provide service health statistics to DNS-based GSLB applications.

This protocol permits a server to roam into any network and maintain the same VIP address. Allowing the server to maintain the same VIP address that is not correlated to the location of the server within a network topology easily enables the use of network layer anycast technology [4]. Also, enabling plug and play SLB configuration will decrease the time, cost, and management required when deploying server farms, simplify anycast deployment, and also minimize the amount of soldier training required to support networks.

This paper will describe the SLB Registration Protocol and an initial Java-based implementation. The paper documents how the SLB manager can be used to provide statistics to an implementation of a DNS-based GSLB. Next, the paper will describe how the SLB and GSLB applications can be used to enable plug and play anycasting. The paper will also describe two anycast Address Resolution Protocol (ARP) implementations to support clients requiring server connections to applications that use dynamic port number assignments (e.g., the H.323 protocol).

## PROTOCOL DESCRIPTION AND OPERATION

A server will initiate a connection to a SLB manager via Transmission Control Protocol (TCP) to register services. The SLB manager will accept registration from the real server and communicate the existence of the server's VIP address to the GSLB application and network infrastructure equipment. When the services supported by the server fails, the SLB and GSLB will remove service availability from the infrastructure.

## REAL SERVER AND SLB MANAGER COMMUNICATIONS

The typical sequence of message exchanges between a real server, the SLB manager, and a GSLB application is shown in Figure 1. First a server will register with the SLB manager. The SLB manager will acknowledge registration of the server. If the server requires a VIP address, the server will request to lease a VIP from the SLB manager. Once the VIP has been provided to the server, the server will install the VIP address as a loopback interface. Next, the server will register its services with the SLB manager.

Once it processes the service registration request, the SLB manager will perform a health check on the service. If successful, the SLB manager programs the SLB to represent the service provided by the server. The SLB

[1] Per MITRE Public Release Memorandum 03-0701 this paper is Approved for Public Release; Distribution Unlimited.

manager will also alert the routing daemon to advertise the assigned VIP address to the network.

At this point, the server will deliver service health status messages to the SLB manager. The SLB manager will also perform its own independent health checks on the registered service. If a service health failure occurs, the SLB manager will automatically remove the server from the server farm. Upon health failure of the last server within a common local server farm, the SLB manager will alert the local routing daemon to remove the VIP address from the routing infrastructure.

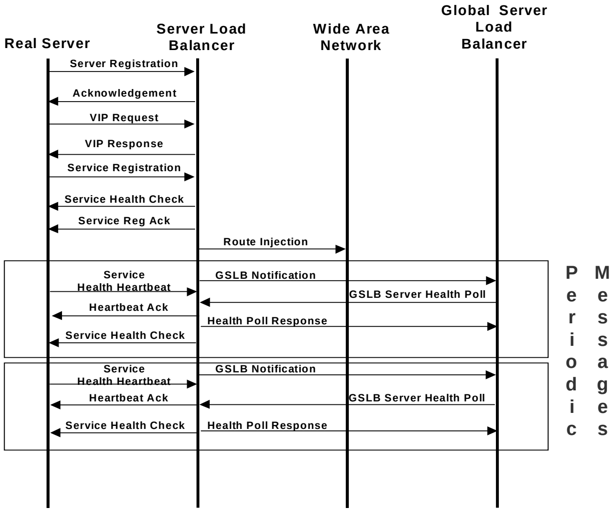

**Figure 1. Server and SLB Manager Message Flows**

## SLB MANAGER AND GSLB APPLICATION COMMUNICATION

The SLB manager will also provide a GSLB application with information about the existing services of its registered real servers. In this case, once the SLB manager completes the registration of a service, it will notify one or more applications of the new service and the new VIP address associated with the service. This notification is transmitted as an IP multicast datagram.

The GSLB is a DNS server that is configured to be authoritative for one or more specific zones. The GSLB will register with its local router to receive the multicast service notifications. Each notification includes the unique IP address of the SLB that generated the service notification as well as the unique DNS name and VIP address required by the service. The GSLB will insert the entry into its DNS database to track what services each SLB maintains. At this point the GSLB will begin to periodically poll the SLB for the information concerning the health of the service.

Upon reception of a poll from the GSLB, the SLB manager will respond with a score that describes the health of the service. This score is normalized and accounts for the number of servers supporting the service and a weighted time average of the number of connections. When the GSLB learns of multiple dispersed VIP addresses that support the same service (i.e., identified by the DNS name), the GSLB will use the information provided by the different SLB managers to influence its DNS responses.

The GSLB application can also be configured to collect other network performance information, such as percent bandwidth utilization for key communication links and round trip time measurements between Tactical Operation Centers (TOCs) and key information servers.

## IMPLEMENTATION

An initial Java-based implementation of the SLB Registration Protocol has been completed. It has been written to work with an implementation of the Open Source Linux Virtual Server (LVS) [5] and with an experimental version of the Cisco IOS that includes an embedded Java Virtual Machine[2].

The SLB Registration Protocol implementation has been designed to permit the SLB manager to easily connect to different SLB implementations and interfaces. A simple diagram describing the components designed to support the SLB Registration Protocol is provided by Figure 2.

The modules highlighted within Figure 2 illustrate the components developed to support the SLB Registration Protocol. The Commercial off the Shelf (COTS) SLB and the real servers shown within Figure 2 represent external components that would interface with the SLB Registration Protocol. For these last 2 modules, the real server is the server being managed and the COTS SLB is the SLB application or appliance being configured by the SLB manager.

[2] This is a pre-production version of Cisco IOS with an embedded Java Virtual Machine. This Cisco IOS is currently not available on any basis for commercial use, however, Cisco is willing to discuss access to pre-release IOS images for research and experimentation.

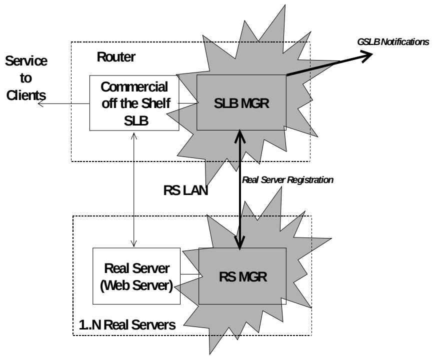

**Figure 2. SLB Registration Protocol Software Components**

## ROUTE INJECTION IMPLEMENTATION

The SLB manager will configure the local OSPF implementation to inject a route to advertise a new VIP address. To support OSPF with the Linux Virtual Server we leveraged the open source Zebra [6] router implementation. The Java-based SLB manager will automatically configure the Zebra OSPF to advertise the appropriate VIP route for the Linux implementation. For the Cisco-based SLB manager implementation, we automatically configure the Cisco to advertise a VIP route via OSPF.

One difference between the LVS and Cisco IOS SLB implementation is associated with how the VIP is represented by the different systems. The LVS SLB implementation represents a VIP address as a secondary IP address on one of the computer's network interface cards. This implementation allows Zebra'a OSPF to advertise the VIP address as an internal OSPF route. Cisco's IOS SLB implementation represents the VIP as a static route to a null interface. This requires the VIP static route to be redistributed into the routing protocol. For OSPF, this means the VIP route is advertised as an external route.

By default, OSPF prefers an internal route when compared to a second route to the same destination that is marked as external. Therefore, when using network layer anycast within a mixed LVS/Zebra and Cisco IOS environment, all LVS SLBs will be used prior to any Cisco IOS routers. OSPF does not permit this behavior to be easily modified. Other routing protocols may be more conducive to avoiding this limitation.

## NETWORK PERFORMANCE INFORMATION AND GSLB

Our GSLB implementation leverages a partial DNS server called Load Balance Named (LBNamed) [7]. We modified LBNamed to accept service registration and performance information from the SLB manager software and to accept network performance information from Cisco routers. The Cisco routers execute a Java program to gather and push network performance information to the GSLB.

This information includes the 5-minute input and output data rates for key communication links. The Java program on the Cisco router will also instruct key TOC WAN edge routers to receive the multicast SLB Manager service notification messages. Once the Cisco router receives this notification message it will configure itself to collect transaction round trip time (RTT) measurements between itself and the new services announced. Obtaining RTT measurement data is enabled through the use of the Cisco Service Assurance Agent (SAA) [8]. This RTT data is provided to the GSLB.

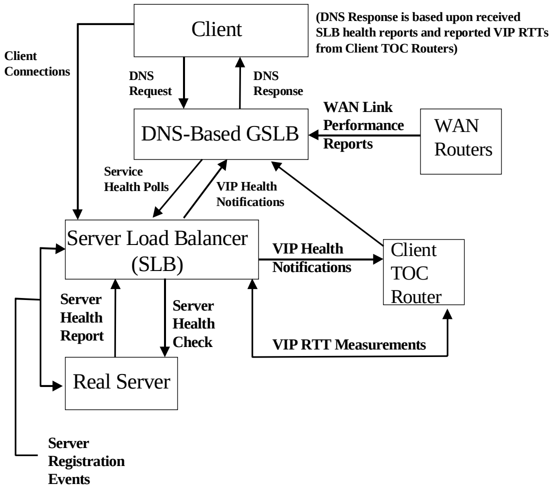

**Figure 3. GSLB Component Interaction**

In addition to obtaining this network data and RTT data from network sources, the GSLB will be populated with data that describes the network topology. In particular, this will include information that maps local DNS server addresses to their supported IP subnetwork addresses and WAN access routers. This data will permit the GSLB to associate network topology information with client and server location. Combining this information with network and server performance measurements permits the GSLB to provide clients with DNS responses that can lead to optimal client-server response times [9]. Figure 3 illustrates the flows between the various network components that affect the GSLB decision process.

## ENABLING ANYCAST

The Server Load Balancing Registration Protocol can be used to automate support for anycast technology. Since the protocol allows distributed servers to register the same VIP address with geographically dispersed SLB Managers, network layer anycast can be easily deployed. Each SLB Manager will inject the anycast VIP address into the OSPF routing tables from their dispersed locations. The OSPF routing protocol will be used to automatically connect clients to the closest server. This simplifies server location issues associated with finding key servers such as DNS and other directory servers.
In addition, this protocol can permit servers to roam into different wireless cells without changing its VIP address. If each wireless cell is associated with a different IP subnet, the server can get a new local IP address via DHCP and still maintain a constant VIP address by registering with a local SLB manager. If necessary, HTTP redirect technology or anycast ARPs can be used to associate a VIP-assigned web server with a server's unique real IP address.

The Server Load Balancing Registration Protocol can also be used to automate support for application layer anycast as well. In this case, separate dispersed SLB managers will assign unique VIPs to geographically dispersed servers providing the same or similar service. Automatic insertion of the VIPs and services into the GSLB database allows the operation of plug and play application layer anycast as well. The GSLB application will then work to direct clients to an optimal server. The VIP addressing, naming and performance information will be discarded by the GSLB upon notification from the SLB manager or lost communications between the GSLB and SLB manager. Additional information on leveraging GSLB concepts for anycast networks with QoS can also be found in [10].

## ANYCAST ARPS

Video Teleconferencing (VTCs) can be supported when H.323 clients register with H.323 Multipoint Control Units (MCUs). Anycast techniques will allow roaming H.323 clients to find the closest MCU. Also, when using the SLB Registration Protocol the MCU server can roam anywhere in the network and register its anycast IP address with a local SLB. However, H.323 signaling techniques allocates voice and video UDP port numbers dynamically. This limits our ability to take advantage of generic server load balancing of MCUs.

In lieu of using load balancing to support MCU H.323 connections, we can leverage network layer anycast ARP to allow the H.323 client to find the unique address of the MCU. The closest SLB manager supporting the anycast IP address will respond to the anycast ARP request with the individual IP addresses of all MCUs locally registered. The list will include an MCU health score for each registered MCU. The H.323 client can pick a preferred MCU from this list and connect to the unique address of the MCU.

With this previous example, the H.323 client must know the IP anycast address of the MCU. If this is not known or possible, an alternative anycast ARP implementation will permit the H.323 client to request a list of MCU anycast addresses directly from the GSLB. In this implementation, the response from the GSLB will include the unique address of the SLB managers that maintain registered MCUs. The GSLB will prioritize the information within the response with a health score associated with each SLB. The client can then pick a preferred SLB manager and address the anycast ARP directly to the SLB manager. The SLB manager will respond with the unique IP addresses of the SLB's directly attached MCUs.

What makes our anycast ARP unique is that the client request includes not only the anycast IP address, but also the requested application protocol (e.g., HTTP or DNS), transport protocol port numbers (e.g., 80, 8080, 53) , as well as the transport protocol type (e.g., either TCP or UDP). In lieu of including this detailed information in the anycast ARP, a wildcard entry will allow the SLB manager or GSLB to include all services and associated health scores for the anycast address within the response.

## CONCLUSION

The SLB Registration Protocol is a simple and efficient way to automate anycast addressing. By leveraging plug and play SLB configuration concepts, we significantly help to reduce complexity associated with server mobility in the battlefield. Through plug and play GSLB we have enabled optimal server selection by using both network and server performance attributes. Our anycast ARP implementation also permits protocols not normally supported by SLB to function very well within our environment.

## REFERENCES

[1] T. Schroeder, S Goddard, B. Ramamurthy, "Scalable Web Server Clustering Technologies," IEEE Network, May/June 2000, pp. 38-45.

[2] E. Zegura, M. Ammar, Z. Fei, S. Bhattacharjee, "Application Layer Anycasting: A Server Selection Architecture and Use in a Replicated Web Service," IEEE/ACM Transactions on Networking, Vol. 8, No. 4, August 2000, pp. 455-467.

[3] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, B. Weihl, "Globally Distributed Content Delivery," IEEE Internet Computing, September/October 2002, pp. 51-58.

[4] Chris Metz, "IP Anycast Point-to-(Any) Point Communication," IEEE Internet Computing, March/April 2002, pp. 94-98.

[5] http://www.linuxvirtualserver.org

[6] http://www.zebra.org

[7] http://www.stanford.edu/~schemers/docs /lbnamed/lbnamed.html/

[8] http://www.cisco.com/en/US/poducts/sw/ iosswrel/ps1830/products_feature_guide09186a0080087a78.html

[9] R. Carter and M. Crovella, "Dynamic Server Selection using Bandwidth Probing in Wide Area Networks," Computer Science Department, Boston University, BU-CS-96-007, March 18, 1996.

[10] Fang Hao and Ellen W. Zegura, "QoS Routing for Anycast Communications: Motivation and an Architecture for DiffServ Networks," IEEE Communications Magazine, June 2002, pp. 48-56.