WHITE PAPER

MICROSOFT LIVE COMMUNICATIONS SERVER 2005 LOAD BALANCING WITH FOUNDRY NETWORKS SERVERIRON PLATFORM

NOTE: Foundry's ServerIron load balancing switches have been certified in Microsoft's load balancing LCS 2005 interoperability labs. Microsoft experts executed a variety of tests against Foundry switches. Contact Foundry Networks for details about the testing, and to learn more about the benefits delivered by ServerIron switches when deployed in a Microsoft application infrastructure.

Foundry ServerIron switches, when deployed in front of the Microsoft Live Communications Server 2005 Enterprise Edition, increase application uptime, maximize server farm utilization, and shield the servers and applications from malicious attacks. The switches receive all client requests, and distribute them efficiently to the "best" server among the available pool. ServerIron switches consider server availability, load, response time, and other user-configured performance metrics when selecting a server for incoming client connections. By performing sophisticated and customizable "health checks" to the servers and the Microsoft Live Communications Server 2005, the ServerIron switches quickly identify resource outages in real time and re-direct client connections to other available servers. Server capacity can be increased or decreased on demand without impacting the applications and client connections. When demand grows, IT engineers can simply "slide" in new server resources and configure the ServerIron switch to use the new servers for client connections.

ServerIron switches are application aware and can inspect many types of application level content to perform intelligent switching of client requests to appropriate servers. Application switching eliminates the need to replicate content and application functions on all servers, and optimizes overall resource utilization, application performance and availability. ServerIron switches support switching based on broad content types including URL, HTTP headers, HTTP cookies, SSL session IDs, and XML tags.

For implementations where session persistence across multiple TCP ports on the same server is a key requirement, the ServerIron switches offer support for the industry's most advanced and easily customizable load balancing interface. In addition, the power of performance delivered by the ServerIron switches ensures that the applications provide the best end-user response time and immense scalability even when enabled for Layer 4 through 7 switching. Using sticky sessions and track-group switching, a group of transactions from a given client are sent to the server originally selected and has the session created when the client first connected. A crucial benefit of using Foundry ServerIron is its ability to ensure the client stays with one real server so all real time information is preserved as the client continues to communicate across several application ports. Another benefit of using the Foundry ServerIron to load balance is its ability to protect server farms and applications from malicious attacks. The ServerIron switches are proven to defeat wire-speed Gigabit rate Denial of Service (DoS) attacks while maintaining peak application performance. Foundry switches also provide high-performance content inspection and filtering for malicious content, including viruses and worms, which are spread through application level messages to cripple the servers and take down applications. Foundry's ServerIron solution provides immediate ROI, and also improves the ROI of Microsoft Live Communications Server 2005. They support significantly higher application traffic and number of users on existing server resources by maximizing utilization. On-demand and unlimited virtual server farm scalability eliminates the need for forklift upgrades, and dramatically improves the ROI on the server infrastructure. Downtime associated with security breaches and scheduled maintenance is eliminated, resulting in improved availability, which results in savings of tens of thousands to millions of dollars a year.

Figure 1. Live Communications Server 2005 Enterprise Edition Logical Configuration.

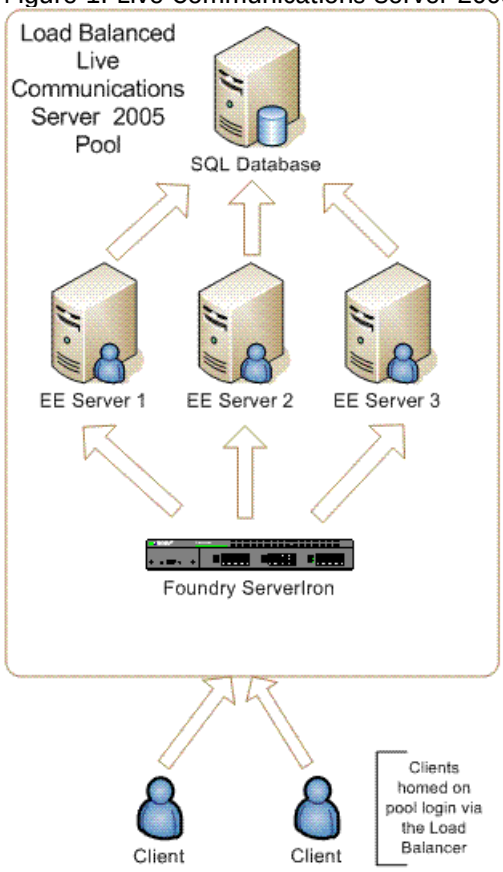
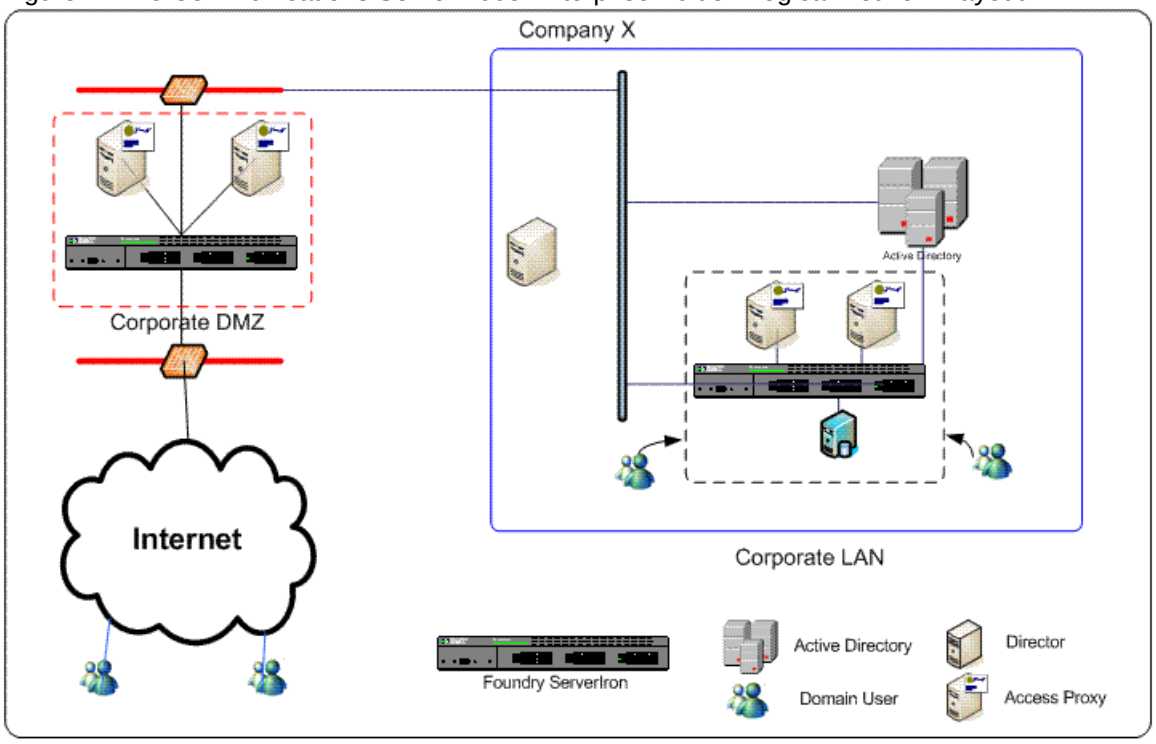


Figure 2. Live Communications Server 2005 Enterprise Edition Logical Network Layout.



Follow these steps to configure ServerIron A for Server load balancing. A second configuration has been added at the end of the paper for customers that wish to build Hot standby Server Load Balancing. In a typical hot standby configuration, one ServerIron is the active device and performs all the Layer 2 switching as well as the Layer 4 SLB switching while the other ServerIron monitors the switching activities and remains in a hot standby role. If the active ServerIron becomes unavailable, the standby ServerIron immediately assumes the unavailable ServerIron's responsibilities. The failover from the unavailable ServerIron to the standby ServerIron happens transparently to users. Both ServerIrons share a common MAC address known to the clients. Therefore, if a failover occurs, the clients still know the ServerIron by the same MAC address. The active sessions running on the clients continue and the clients and routers do not need to re-ARP for the ServerIron MAC address.

Note: All real servers are connected directly to the ServerIrons with Active Standby NIC configuration where the Active Nic is connected to the active ServerIron.

Enterprise Edition Pool:

The Enterprise Edition of the Live Communications Server 2005 introduces a couple of new terms – Enterprise Edition (EE) Servers and Pool. An EE server provides connection termination, security, authorization, protocol processing and applications. Multiple EEs communicate with a single back-end SQL Server. 'Pool' is the term used to describe this collection of multiple EEs tied to a single backend. Users are now homed on Pools. This allows users to login using any EE

server in a pool. Pools allow flexibility in increasing the capacity of the service by easily adding more EE server. Failure of an EE should not affect service availability.

You can manage a Foundry device using the following interfaces:

- Command Line Interface (CLI) – a text-based interface accessible through a direct serial connection or a Telnet session.
- Web management interface – A GUI-based management interface accessible through an HTTP (web browser) connection.

To manage the ServerIron via CLI:

- At the opening CLI prompt, enter **enable**.
  `ServerIron> enable`
- Access the configuration level of the CLI by entering the following command:
  `ServerIron# configure terminal`
  `ServerIron(config)#`
- To Exit from the configuration level of the CLI, enter the following command:
  `ServerIron(config)# end`
- To save the configuration to NVRAM, enter the following command:
  `ServerIron# write memory`

Add routing:

```
ServerIronA(config)# vlan 1
ServerIronA(config-vlan-1)# router-interface VE 1
ServerIronA(config-vlan-1)# interface VE 1
ServerIronA(config-vlan-1)# ip standby-address 10.0.0.5 255.255.255.0
ServerIronA(config)# ip forward
```

The **ip standby-address** command allows the address to be shared by the two ServerIrons so that the standby ServerIron can provide service for the address if the active ServerIron becomes unavailable.

Set up the default server ports used for SIP:

```
ServerIronA(config)# server port 5060
ServerIronA(config)# tcp
ServerIronA(config)# server port 5061
ServerIronA(config)# tcp
ServerIronA(config)# server port 135
ServerIronA(config)# tcp
```

Real Server setup:

```
ServerIronA(config)# server real EEServer1 10.0.0.10
ServerIronA(config)# port 5060
ServerIronA(config)# port 5061
ServerIronA(config)# port 135
ServerIronA(config)# server real EEServer2 10.0.0.20
ServerIronA(config)# port 5060
ServerIronA(config)# port 5061
ServerIronA(config)# port 135
```

Virtual Server Setup:

```
ServerIronA(config)# server virtual EEPool1 10.0.0.100
ServerIronA(config)# port 5060 sticky
ServerIronA(config)# port 5061 sticky
ServerIronA(config)# port 135 sticky
ServerIronA(config)# bind 5060 EEServer1 5060 EEServer2 5060
ServerIronA(config)# bind 5061 EEServer1 5061 EEServer2 5061
ServerIronA(config)# bind 135 EEServer1 135 EEServer2 135
ServerIronA(config)# track-group 5061 135
```

Access Proxy Array:

The Live Communications Server product allows peering between enterprises using the same server product. This feature is enabled using a proxy server – called an Access Proxy, on the perimeter network boundary. The Access proxy is used by outside users (employees of an enterprise that access the network remotely) and for interconnecting with other enterprises to exchange instant messages and presence information. An Access Proxy typically has 2 NICs, one for the internal and the other for the external network.

Add routing:

```
ServerIronA(config)# vlan 1
ServerIronA(config-vlan-1)# interface VE 1
ServerIronA(config-vlan-1)# ip standby-address 10.1.0.5 255.255.255.0
```

Set up the default server ports used for SIP:

```
ServerIronA(config)# server port 5060
ServerIronA(config)# tcp
ServerIronA(config)# server port 5061
ServerIronA(config)# tcp
```

Real Server setup:

```
ServerIronA(config)# server real APServer1_FromInternal 10.1.0.10
ServerIronA(config)# port 5060
ServerIronA(config)# port 5061
ServerIronA(config)# server real APServer1_FromExternal 10.1.0.11
ServerIronA(config)# port 5060
ServerIronA(config)# port 5061

ServerIronA(config)# server real APServer2_FromInternal 10.1.0.20
ServerIronA(config)# port 5060
ServerIronA(config)# port 5061
ServerIronA(config)# server real APServer2_FromExternal 10.1.0.21
ServerIronA(config)# port 5060
ServerIronA(config)# port 5061
```

Virtual Server Setup:

```
ServerIronA(config)# server virtual APBank1_FromInternal 10.1.0.100
ServerIronA(config)# port 5060 sticky
ServerIronA(config)# port 5061 sticky
ServerIronA(config)# bind 5060 APServer1_FromInternal 5060
APServer2_FromInternal 5060
ServerIronA(config)# bind 5061 APServer1_FromInternal 5061
APServer2_FromInternal 5061
ServerIronA(config)# track-group 5060 5061

ServerIronA(config)# server virtual APBank1_FromExternal 10.1.0.101
ServerIronA(config)# port 5060 sticky
ServerIronA(config)# port 5061 sticky
ServerIronA(config)# bind 5060 APServer1_FromExternal 5060
APServer2_FromExternal 5060
ServerIronA(config)# bind 5061 APServer1_FromExternal 5061
APServer2_FromExternal 5061
ServerIronA(config)# track-group 5060 5061
```

Director Bank:

A Director is a Live Communications Server that has no homed users and provides additional security for the internal network and directs client login requests to the appropriate pool or server within the enterprise (assuming that the enterprise has deployed more than one pool/server). The Director authenticates and authorizes external SIP traffic coming from an Access Proxy.

Add routing:

```
ServerIronA(config)# vlan 1
ServerIronA(config-vlan-1)# interface VE 1
ServerIronA(config-vlan-1)# ip standby-address 10.2.0.5 255.255.255.0
```

Set up the default server ports used for SIP:

```
ServerIronA(config)# server port 5060
ServerIronA(config)# tcp
ServerIronA(config)# server port 5061
ServerIronA(config)# tcp
```

Real Server setup:

```
ServerIronA(config)# server real DrServer1 10.2.0.10
ServerIronA(config)# port 5060
ServerIronA(config)# port 5061
ServerIronA(config)# server real DrServer2 10.2.0.20
ServerIronA(config)# port 5060
ServerIronA(config)# port 5061
```

Virtual Server Setup:

```
ServerIronA(config)# server virtual DrBank1 10.2.0.100
ServerIronA(config)# port 5060 sticky
ServerIronA(config)# port 5061 sticky
ServerIronA(config)# bind 5060 DrServer1 5060 DrServer2 5060
ServerIronA(config)# bind 5061 DrServer1 5061 DrServer2 5061
ServerIronA(config)# track-group 5060 5061
```

Standalone ServerIron Configuration is now completed. Below are the commands for ServerIron B and the steps to complete an Active/ Standby configuration. Foundry Networks, Inc. recommends using the most current Ironware OS to ensure the best possible performance of the Foundry products.

Configure port 13 as the hot standby port on both ServerIrons:

```
ServerIronA(config)# vlan 2
ServerIronA(config-vlan-2)# untag ethernet 13
ServerIronA(config-vlan-2)# no spanning-tree
ServerIronA(config-vlan-2)# exit
ServerIronA(config)# server router-ports 10
ServerIronA(config)# server backup ethernet 13 00e0.5201.0c72 2
```

```
ServerIronB(config)# vlan 2
ServerIronB(config-vlan-2)# untag ethernet 4
ServerIronB(config-vlan-2)# no spanning-tree
ServerIronB(config-vlan-2)# exit
ServerIronB(config)# server router-ports 10
ServerIronB(config)# server backup ethernet 4 00e0.5201.0c72 2
```

The **server router-ports** command identifies the port that is connected to the router. This command is required in hot-standby configurations.

The **server backup** command designates port 13 as the hot standby port and specifies the MAC address the port is backing up.

To configure a ServerIron in an active-standby pair to always be the active ServerIron, enter the following command at the global CONFIG level of the CLI:

```
ServerIronA(config)# server backup-preference 5
```

Enterprise Edition Pool:

Add routing:

```
ServerIronB(config)# vlan 1
ServerIronB(config-vlan-1)# router-interface VE 1
ServerIronB(config-vlan-1)# interface VE 1
ServerIronB(config-vlan-1)# ip standby-address 10.0.0.5 255.255.255.0
ServerIronB(config)# ip forward
```

Set up the default server ports used for SIP:

```
ServerIronB(config)# server port 5060
ServerIronB(config)# tcp
ServerIronB(config)# server port 5061
ServerIronB(config)# tcp
ServerIronB(config)# server port 135
ServerIronB(config)# tcp
```

Real Server setup:

```
ServerIronB(config)# server real EEServer1 10.0.0.10
ServerIronB(config)# port 5060
ServerIronB(config)# port 5061
ServerIronB(config)# port 135
ServerIronB(config)# server real EEServer2 10.0.0.20
ServerIronB(config)# port 5060
ServerIronB(config)# port 5061
ServerIronB(config)# port 135
```

Virtual Server Setup:

```
ServerIronB(config)# server virtual EEPool1 10.0.0.100
ServerIronB(config)# port 5060 sticky
ServerIronB(config)# port 5061 sticky
ServerIronB(config)# port 135 sticky
ServerIronB(config)# bind 5060 EEServer1 5060 EEServer2 5060
ServerIronB(config)# bind 5061 EEServer1 5061 EEServer2 5061
ServerIronB(config)# bind 135 EEServer1 135 EEServer2 135
ServerIronB(config)# track-group 5061 135
```

Access Proxy Array:

Add routing:

```
ServerIronB(config)#                        vlan                        1
ServerIronB(config-vlan-1)#           interface           VE            1
ServerIronB(config-vlan-1)# ip standby-address 10.1.0.5 255.255.255.0
```

Set up the default server ports used for SIP:

```
ServerIronB(config)# server port 5060
ServerIronB(config)# tcp
ServerIronB(config)# server port 5061
ServerIronB(config)# tcp
```

Real Server setup:

```
ServerIronB(config)# server real APServer1_FromInternal 10.1.0.10
ServerIronB(config)# port 5060
ServerIronB(config)# port 5061
ServerIronB(config)# server real APServer1_FromExternal 10.1.0.11
ServerIronB(config)# port 5060
ServerIronB(config)# port 5061

ServerIronB(config)# server real APServer2_FromInternal 10.1.0.20
ServerIronB(config)# port 5060
ServerIronB(config)# port 5061
ServerIronB(config)# server real APServer2_FromExternal 10.1.0.21
ServerIronB(config)# port 5060
ServerIronB(config)# port 5061
```

Virtual Server Setup:

```
ServerIronB(config)# server virtual APBank1_FromInternal 10.1.0.100
ServerIronB(config)# port 5060 sticky
ServerIronB(config)# port 5061 sticky
ServerIronB(config)# bind 5060 APServer1_FromInternal 5060
APServer2_FromInternal 5060
ServerIronB(config)# bind 5061 APServer1_FromInternal 5061
APServer2_FromInternal 5061
ServerIronB(config)# track-group 5060 5061

ServerIronB(config)# server virtual APBank1_FromExternal 10.1.0.101
ServerIronB(config)# port 5060 sticky
ServerIronB(config)# port 5061 sticky
ServerIronB(config)# bind 5060 APServer1_FromExternal 5060
APServer2_FromExternal 5060
ServerIronB(config)# bind 5061 APServer1_FromExternal 5061
APServer2_FromExternal 5061
ServerIronB(config)# track-group 5060 5061
```

Director Bank:

Add routing:

```
ServerIronB(config)# vlan 1
ServerIronB(config-vlan-1)# interface VE 1
ServerIronB(config-vlan-1)# ip standby-address 10.2.0.5 255.255.255.0
```

Set up the default server ports used for SIP:

```
ServerIronB(config)# server port 5060
ServerIronB(config)# tcp
ServerIronB(config)# server port 5061
ServerIronB(config)# tcp
```

Real Server setup:

```
ServerIronB(config)# server real DrServer1 10.0.0.10
ServerIronB(config)# port 5060
ServerIronB(config)# port 5061
ServerIronB(config)# server real DrServer2 10.0.0.20
ServerIronB(config)# port 5060
ServerIronB(config)# port 5061
```

Virtual Server Setup:

```
ServerIronB(config)# server virtual DrBank1 10.2.0.100
ServerIronB(config)# port 5060 sticky
ServerIronB(config)# port 5061 sticky
ServerIronB(config)# bind 5060 DrServer1 5060 DrServer2 5060
ServerIronB(config)# bind 5061 DrServer1 5061 DrServer2 5061
ServerIronB(config)# track-group 5060 5061
```

WHITE PAPER
MICROSOFT LIVE COMMUNICATIONS SERVER 2005
LOAD BALANCING WITH FOUNDRY NETWORKS
SERVERIRON PLATFORM

Bryan Boretsky
Systems Engineer

Foundry Networks, Inc.
Headquarters
2100 Gold Street
P.O. Box 649100
San Jose, CA 95164-9100