MICROSOFT
TRAINING
AND CERTIFICATION

Microsoft Official
Curriculum

Module 8: Concepts of A Network Load Balancing Cluster

Contents

Microsoft®

Program Manager: Don Thompson
Product Manager: Greg Bulette
Instructional Designers: April Andrien, Priscilla Johnston, Diana Jahrling
Subject Matter Experts: Jack Creasey, Jeff Johnson
Technical Contributor: James Cochran
Classroom Automation: Lorrin Smith-Bates
Graphic Designer: Andrea Heuston (Artitudes Layout & Design)
Editing Manager: Lynette Skinner
Editor: Elizabeth Reese
Copy Editor: Bill Jones (S&T Consulting)
Production Manager: Miracle Davis
Build Manager: Julie Challenger
Print Production: Irene Barnett (S&T Consulting)
CD Production: Eric Wagoner
Test Manager: Eric R. Myers
Test Lead: Robertson Lee (Volt Technical)
Creative Director: David Mahlmann
Media Consultation: Scott Serna
Illustration: Andrea Heuston (Artitudes Layout & Design)
Localization Manager: Rick Terek
Operations Coordinator: John Williams
Manufacturing Support: Laura King; Kathy Hershey
Lead Product Manager, Release Management: Bo Galford
Lead Technology Manager: Sid Benavente
Lead Product Manager, Content Development: Ken Rosen
Group Manager, Courseware Infrastructure: David Bramble
Group Product Manager, Content Development: Julie Truax
Director, Training & Certification Courseware Development: Dean Murray
General Manager: Robert Stewart

Instructor Notes

Presentation:
90 Minutes

Lab:
45 Minutes

This module provides students with an overview of Network Load Balancing concepts. The module begins by comparing various load balancing technologies and identifies the applications and services that benefit from a clustering solution. The students are then introduced to the functionality and configuration of the Network Load Balancing driver.

After completing this module, students will be able to:

- Describe the concepts of the Network Load Balancing solution.
- Describe the application and services configuration for Network Load Balancing hosts.
- Describe the functionality of the Network Load Balancing driver.
- Identify the components for the Network Load Balancing driver architecture.

Materials and Preparation

This section provides the materials and preparation tasks that you need to teach this module.

Required Materials

To teach this module, you need Microsoft® PowerPoint® file 2087A_08.ppt.

Preparation Tasks

To prepare for this module, you should:

- Read all of the materials for this module.
- Complete Lab A: Planning an Installation.
- Study the review questions and prepare alternative answers to discuss.
- Anticipate questions that students may ask. Write out the questions and provide the answers.
- Be familiar with all of the clustering technologies discussed and be able to discuss round robin DNS and compare it to the Microsoft clustering technology solutions.
- Be familiar with the concepts of client and session state and be able to discuss them in the context of a Network Load Balancing solution.
- Be very familiar with the functionality of the Network Load Balancing driver and how it manages and balances Internet Protocol (IP) traffic.
- Be able to discuss cluster convergence.
- Be able to discuss the concepts of scalability and high availability in the context of a Network Load Balancing cluster.

- Be able to discuss the filtering algorithm.
- Be able to discuss all of the components of the Network Load Balancing driver.
- Be able to discuss the IP transmission modes.
- Be able to discuss the functionality of the primary and dedicated IP addresses.
- Be able to describe the port rules parameters for the Network Load Balancing driver.

Module Strategy

Use the following strategy to present this module:

- Network Load Balancing Concepts

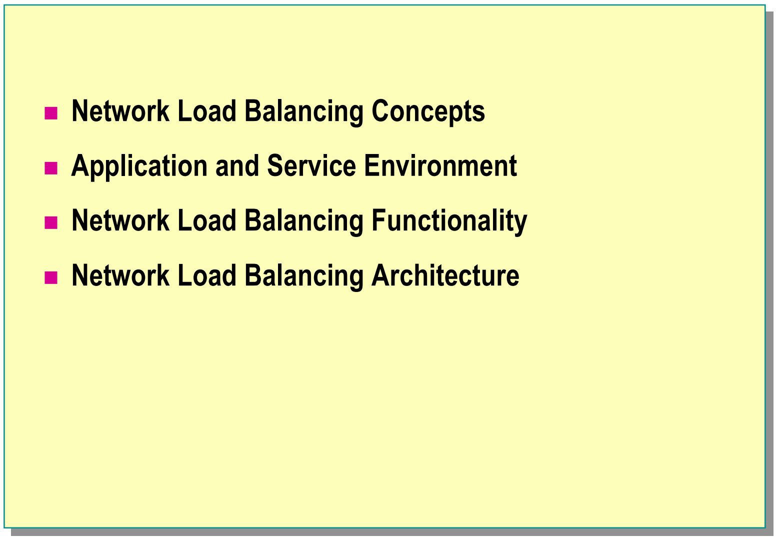
  This topic is an overview of Network Load Balancing concepts.

  - Discuss the various clustering technologies and how they compare to Network Load Balancing.
  - Briefly review the features of Network Load Balancing.
  - Emphasize that there is no single point of failure with Network Load Balancing.
  - Compare other load balancing solutions to Network Load Balancing by using the graphic.
  - Demonstrate the operations of a Network Load Balancing cluster by using the graphic.
  - Demonstrate the concepts of balancing client connections by using the graphic.
  - Carefully explain the concept of high availability by using the graphic.
- Application and Service Environment
  - Identify the applications and services environment and discuss the two kinds of client state and how they are managed.
- Network Load Balancing Functionality
  - Emphasize how the Network Load Balancing driver balances client connections and supports multiple client connections by using the graphics.
  - Discuss the concept of cluster convergence.
  - Explain the dynamics of high availability within a Network Load Balancing cluster by using the graphic.
  - Explain the scalability concepts within a Network Load Balancing cluster by using the graphics.
- Network Load Balancing Architecture
  - Demonstrate the logical position of the Network Load Balancing driver within the Transmission Control Protocol/Internet Protocol (TCP/IP) stack by using the graphic.
  - Emphasize the importance of properly configuring the Network Load Balancing driver and selecting the appropriate IP transmission modes.
  - Ensure that the students understand the unicast and multicast modes.
  - Emphasize the importance of setting consistent port rules for the Network Load Balancing cluster hosts.
  - Discuss the distribution of the incoming client connections based on affinity.

Overview

Topic Objective
To provide an overview of the module topics and objectives.

Lead-in
In this module, you will learn about the features and functions for Microsoft Windows 2000 Network Load Balancing service and how you can use a clustering solution to improve availability, scalability, and load balancing.



- **Network Load Balancing Concepts**
- **Application and Service Environment**
- **Network Load Balancing Functionality**
- **Network Load Balancing Architecture**

****************************ILLEGAL FOR NON-TRAINER USE*****************************

Microsoft® Windows® 2000 Advanced Server and Microsoft Windows 2000 Datacenter Server operating systems include two clustering technologies; Cluster service and Network Load Balancing service.

Cluster service is intended primarily to provide failover support for critical line-of-business applications, such as databases, messaging systems, and file/print services. Network Load Balancing service balances incoming Internet Protocol (IP) traffic among multihost clusters. This module will address Network Load Balancing service in detail.

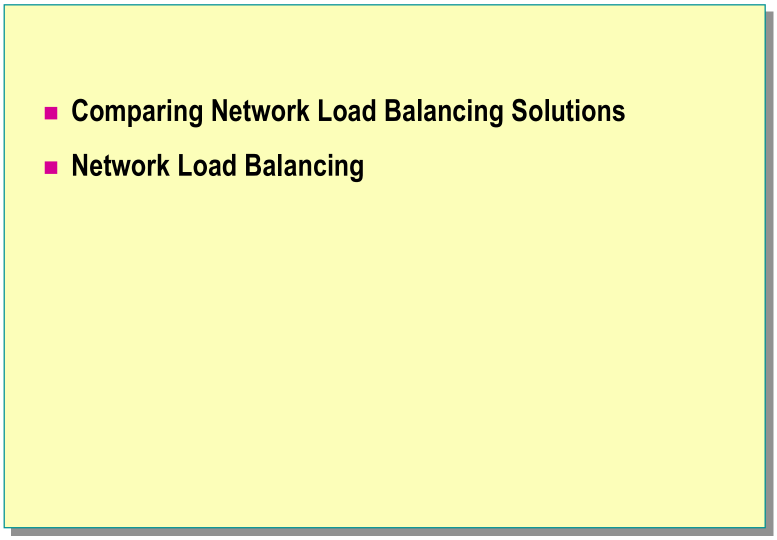
After completing this module, you will be able to:

- Describe the concepts of the Network Load Balancing solution.
- Describe the application and services configuration for Network Load Balancing hosts.
- Describe the functionality of the Network Load Balancing driver.
- Identify the components for the Network Load Balancing driver architecture.

◆ Network Load Balancing Concepts

Topic Objective
To give an overview of Network Load Balancing concepts.

Lead-in
Internet server programs supporting mission-critical applications and services, such as financial transactions, database access, corporate intranets, and other key functions, must run 24 hours a day, seven days a week.



- **Comparing Network Load Balancing Solutions**
- **Network Load Balancing**

******************************ILLEGAL FOR NON-TRAINER USE********************************

Delivery Tip
Before leaving this page, ask the students what load balancing solutions they have implemented. Emphasize to the students that there are no single points of failure with Network Load Balancing.

Internet server programs supporting mission-critical applications and services, such as financial transactions, database access, corporate intranets, and other key functions must run 24 hours a days, seven days a week. In addition, network applications and servers need the ability to scale performance to handle large volumes of client requests without creating unwanted delays.

Network load balanced clusters enable you to manage a group of independent servers as a single system for higher availability, easier manageability, and greater scalability.

You can use Network Load Balancing service to implement enterprise-wide highly available and scalable solutions for the delivery of Transmission Control Protocol/Internet Protocol (TCP/IP) based services and applications.

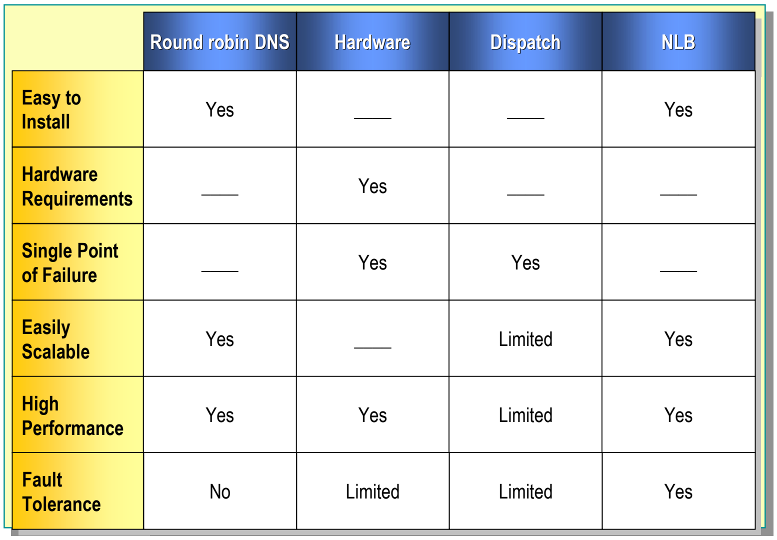
Network Load Balancing has many advantages over other load balancing solutions that can introduce single points of failure or performance bottlenecks. Because there are no special hardware requirements for Network Load Balancing service, you can use any industry standard compatible computer in a Network Load Balancing cluster.

Important The Network Load Balancing driver requires that TCP/IP be installed and supports only Ethernet or Gigabit Ethernet network adapters. Network Load Balancing does not support network basic input/output system (NetBIOS) Enhanced User Interface (NetBEUI) or Internetwork Packet Exchange (IPX).

Comparing Network Load Balancing Solutions

Topic Objective
To compare load balancing solutions.

Lead-in
Comparing load-balancing solutions will enable you to determine the advantages and disadvantages of each and to implement a solution that will provide ease of installation, avoid specialized hardware, and single points of failure.



	Round robin DNS	Hardware	Dispatch	NLB
Easy to Install	Yes	—	—	Yes
Hardware Requirements	—	Yes	—	—
Single Point of Failure	—	Yes	Yes	—
Easily Scalable	Yes	—	Limited	Yes
High Performance	Yes	Yes	Limited	Yes
Fault Tolerance	No	Limited	Limited	Yes

Delivery Tip
This topic is an overview of alternatives to Network Load Balancing, review each of the technologies and point out the benefits of Network Load Balancing.

Key Point
The table illustrates that the alternative solutions to Network Load Balancing have limitations in some categories. These limitations are due to the single point of failure, packet translation, and limited communication between the hosts in a cluster when implementing these solutions.

Comparing load balancing solutions will enable you to determine the advantages and disadvantages of each and to implement a solution that will provide ease of installation, avoid specialized hardware, and avoid single points of failure.

Network Load Balancing service is a high performance and cost-effective solution for both load balancing and fault tolerance where services and applications use Windows 2000-based computers.

However, selection of a viable solution for your enterprise can depend on many factors, including the operating system in use, current network hardware, and network types.

Load balanced clients are able to access a pool of servers with other load balancing solutions, such as round robin DNS, hardware-based load balancing and dispatcher software load balancing.

Round Robin DNS

Round robin DNS is a common solution for enabling a limited, static form of TCP/IP load balancing for Internet server farms.

Consider the following example in which there are three IP address entries for the same host name on a DNS server.

- MyRRDNSWeb IN A 172.17.21.31
- MyRRDNSWeb IN A 172.17.21.35
- MyRRDNSWeb IN A 172.17.28.41

Using the previous list of round robin DNS IP address entries, when a client sends a query, the DNS server returns all three IP addresses to the DNS client, but typically the client uses only the first one in the list. The next time the DNS server receives a query for this host the order of the list is changed in a cyclic permutation or round-robin, meaning that the address that was first in the previous list is now last in the new list. So if a client chooses the first IP address in the list, it now connects to a different server. In the event of a server failure, round robin DNS will continue to route requests to the failed server until you manually remove the SRV (service) resource record from DNS.

Hardware-Based Load Balancing

Hardware-based load balancing directs client requests for a single IP address to multiple hosts within a cluster. Hardware load balancers typically use a technique called network address translation (NAT), which exposes one or more virtual IP address to clients and forwards data for the designated hosts by translating IP addresses and resending network packets. This technique introduces a single point of failure, the computer performing the redirection of packets, between the cluster and the clients. To achieve high availability with this solution, you need a backup load balancer.

Dispatcher Software Load Balancing

This load balancing solution requires one dispatch server to handle all incoming connection requests, where they are then retransmitted to other servers in the network. This solution limits throughput and restricts performance because the entire cluster’s throughput is limited by the speed and processing power of the dispatch server. The single dispatch server represents a single point of failure, which must be eliminated by moving the dispatching function to a second computer after a failure occurs.

Network Load Balancing

Network load balancing is a fully distributed, software-based solution and does not require any specialized hardware or network components. Network load balancing does not require a centralized dispatcher because all hosts receive inbound packets, and redundancy is provided according to the number of hosts within the cluster.

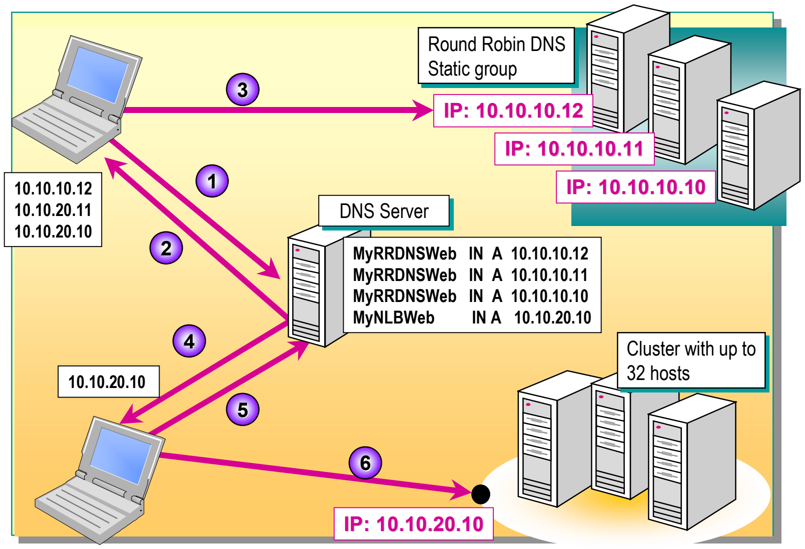
The filtering algorithm for network load balancing is much more efficient in its packet handling than centralized load balancing programs, which must modify and retransmit packets. Network load balancing provides a much higher aggregate bandwidth on similar network configurations.

Note The slide shows that alternative solutions to network load balancing have limitations in some categories. These limitations are due to the single point of failure, packet translation, and limited communication between the hosts in a cluster.

Network Load Balancing

Topic Objective
To introduce the configuration performance and management features of Network Load Balancing.

Lead-in
Many enterprise solutions must address client access to services and applications based on connections to selected TCP/IP addresses, protocols, and port numbers.


Round Robin DNS Static group
IP: 10.10.10.12
IP: 10.10.10.11
IP: 10.10.10.10
3
1
2
10.10.10.12
10.10.20.11
10.10.20.10
DNS Server
MyRRDNSWeb IN A 10.10.10.12
MyRRDNSWeb IN A 10.10.10.11
MyRRDNSWeb IN A 10.10.10.10
MyNLBWeb IN A 10.10.20.10
4
5
6
10.10.20.10
Cluster with up to 32 hosts
IP: 10.10.20.10

Many enterprise solutions must address client access to services and applications that are based on connections to selected TCP/IP addresses, protocols, and port numbers. For example, Internet Information Services (IIS) provides service to clients on IP (TCP, 80). If this single IP host were to fail or become overloaded, client access to the service or application may be prevented or fall below a designated performance level.

Configuring multiple hosts to increase availability, scalability, and fault tolerance for applications and services is one solution. However, this solution may involve specialized network hardware, complex network configuration, and management of individual hosts. For example, multiple hosts functioning as Web servers, each with an individual IP address, could be resolved by multiple entries in round robin DNS. As shown in the graphic where the arrows 1, 2, 3 represent a client Uniform Resource Locator (URL) query, DNS response and cluster connection request. Each server is independent and should a server fail, the static load balancing provided by round robin DNS may prevent clients from accessing their Web application.

To resolve client connection problems, Windows 2000 Network Load Balancing service allows multiple computers or hosts, configured in a logical group called a network load balancing cluster, to respond to client connection requests made to a single virtual IP address. For example, as shown in the graphic, arrows 4, 5, 6 represent a client URL query, DNS response, and a cluster connection request. You will notice that only one IP address is supplied to the client.

Network Load Balancing Features

Windows 2000 Network Load Balancing service provides the following configuration, performance, and management features:

- *TCP/IP support*. Services and applications can be delivered to the client by using specified TCP/IP protocols and ports that can take advantage of network load balancing.
- *Load balancing*. Incoming client connections are load balanced among cluster members based on a distributed algorithm that the Network Load Balancing service executes and rules that you have configured for the cluster.
- *High availability*. Detects the failure of a host within the cluster, and within seconds dynamically reconfigures and redistributes subsequent client requests to hosts that are still viable members of the cluster.
- *Scalability*. Removes or adds hosts without shutting down the cluster; the maximum number of hosts that you can add within a cluster is 32 hosts.
- *Remote Manageability*. Allows remote control of the cluster from any Windows 2000 or Microsoft Windows NT® system.

Network Load Balancing Driver

The Network Load Balancing service is a driver, Wlbs.sys, which you must load on each member server, or host, in the cluster. Wlbs.sys includes a statistical mapping algorithm that the cluster hosts collectively use to determine which host handles each incoming request.

You install the driver on each of the cluster hosts, and you configure the cluster to present a virtual IP address to client requests. The client requests go to all of the hosts in the cluster, but only the mapped host accepts and handles the request. All of the other hosts in the cluster drop the request.

Network Load Balancing Cluster Configuration

After you install the driver, you must configure it before the host can join a cluster. You must configure three groups of information on each host: cluster parameters, host parameters, and port rules, before it is possible to create or join a cluster. Configuring the driver allows you to:

- Select the cluster virtual IP address option.
- Customize the cluster according to the various hosts' capacities and sources of client requests.
- Specify that one host handles all of the client requests with the others serving as failover alternatives.
- Divide the load of incoming client requests among the hosts evenly or according to a specified load partitioning weight.

Network Load Balancing Service Management

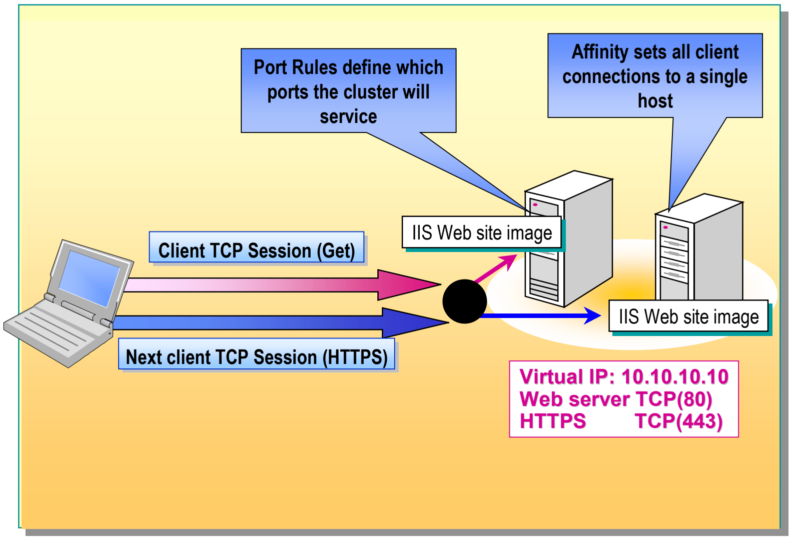
An administrator controls Network Load Balancing service by using the command line utility, Wlbs.exe, which permits interactive and scripted management of a cluster. You can use Wlbs.exe both locally and remotely to control and administer a cluster and the member hosts. With Wlbs.exe you can:

- Examine the status of a running network load balancing cluster.
- Start and stop all or individual hosts in a network load balancing cluster.
- Enable and disable rule processing for specified rules (port numbers).

◆ Application and Service Environment

Topic Objective
To describe the application and services configuration for Network Load Balancing hosts.

Lead-in
When a Web server application maintains state information about a client session across multiple TCP connections, it is important that all TCP connections for this client be directed to the same cluster host.


Port Rules define which ports the cluster will service
Affinity sets all client connections to a single host
IIS Web site image
Client TCP Session (Get)
IIS Web site image
Next client TCP Session (HTTPS)
Virtual IP: 10.10.10.10
Web server TCP(80)
HTTPS TCP(443)

Delivery Tips
Ensure that the students understand how the Network Load Balancing driver can be configured to manage application data state and session state.

When a Web server application maintains state information about a client session across multiple TCP connections, it is important that all TCP connections for the client are directed to the same cluster host.

Network Load Balancing service can load balance any application or service that uses TCP/IP as its network protocol and is associated with a specific TCP or User Datagram Protocol (UDP) port. The distributed algorithm that is used to determine which host responds to a TCP connection request or incoming UDP packet can include the port number in the decision. Including the port number in the decision means that for any client, different members of the Network Load Balancing cluster may service connection requests or packets addressed to different port numbers on the virtual IP address.

Tip While configuring a network load balancing cluster, you need to consider the type of application or service that the virtual server is providing, and select the appropriate configuration for network load balancing hosts.

Port Rules

The Network Load Balancing driver uses port rules that describe which traffic to load balance and which traffic to ignore. By default, the Network Load Balancing driver configures all ports for load balancing. You can modify the configuration of the Network Load Balancing driver that determines how incoming network traffic is load balanced on a per-port basis by creating port rules for each group of ports or individual ports as required. Each port rule configures load balancing for client requests that use the port or ports covered by the port range parameter. How you load-balance your applications is mostly defined by how you add or modify port rules, which you create on each host for any particular port range.

Client State

To configure a Network Load Balancing cluster to correctly handle clients and applications, which maintain state information, requires you to select appropriate settings for port rules and affinity.

Discussion of Network Load Balancing clusters requires clarification of two types of client states, *application data state* and *session state*:

- *Application data state*. It is important to consider whether the server application makes changes to a data store and whether the changes are synchronized across instances of the application (the instances that are running on the network load balancing cluster hosts).

  An example of an application that does not make changes to a data store is a static Web page that an IIS server supports. An example where the application synchronizes data store changes is the use of an Active Server Pages (ASP) based e-commerce site where client state information (their shopping basket contents) is stored in a database, which all members of the network load balancing cluster can access.

- *Session state*. The session state refers to client data that is visible to a client request for the duration of a session. Session state can span multiple TCP connections, which can be either simultaneous or sequential.

  An example of an application that uses this type of state is a Web site that uses server-side cookies to record user navigation. An example of an application that does not use this type of state is a Web site that stores the client navigation information in a client-side cookie, which allows use of the navigation information to any member of a network load balancing cluster servicing the request.

Affinity

Network Load Balancing assists in preserving session state through client affinity settings for each port rule that Network Load Balancing creates. These settings direct all TCP connections from a given client address or class of client addresses to the same cluster host. Directing the connections to the same cluster host allows the server applications in the designated host memory to correctly maintain the session state.

Server-Side Applications and Services

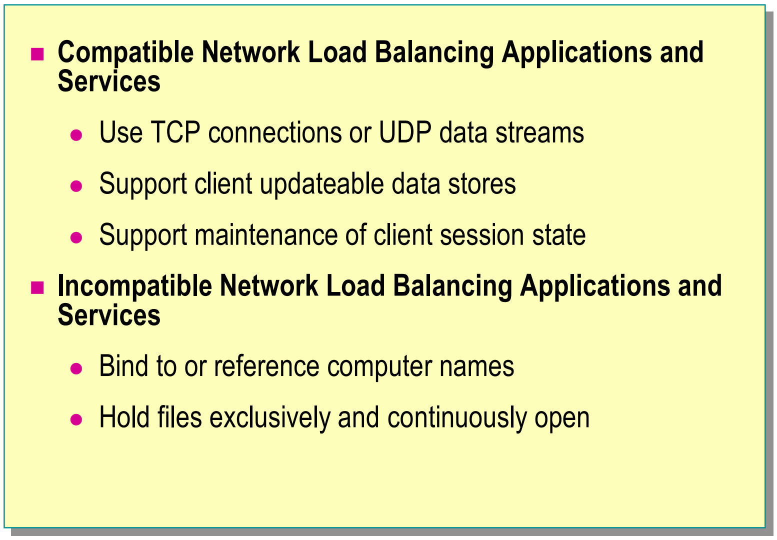
You do not need to modify server applications and services to take advantage of load balancing. However, the system administrator needs to install the applications on each host and ensure that any required synchronization and state issues are addressed. The administrator starts load-balanced applications on all cluster hosts by enabling or disabling port rules for the cluster virtual IP address.

The Network Load Balancing service does not directly monitor server applications, such as a Web server, for continuous and correct operation, so it is recommended that you monitor complex applications and services running over multiple servers.

Applications and Services

Topic Objective
To define the characteristics of Network Load Balancing applications and services.

Lead-in
As Web-based applications continue to gain in importance, it is necessary to host these applications on a flexible platform that provides scalability, reliability, and availability.



- **Compatible Network Load Balancing Applications and Services**
  - Use TCP connections or UDP data streams
  - Support client updateable data stores
  - Support maintenance of client session state
- **Incompatible Network Load Balancing Applications and Services**
  - Bind to or reference computer names
  - Hold files exclusively and continuously open

As Web-based applications continue to become more important, it is necessary to host these applications on a flexible platform that provides scalability, reliability, and availability.

You can satisfy application performance requirements by deploying applications with the following characteristics on a network load balancing infrastructure.

Applications

Applications must have the following characteristics to work with network load balancing:

- They must use TCP connections or UDP data streams.
- If client data changes, you must design applications to provide a means of synchronizing updates to client data that is shared on multiple instances across the cluster.
- If session state is important, applications must use the appropriate affinity setting or provide a means (such as a client cookie or reference to a back-end database) of maintaining session state in order to be uniformly accessible across the cluster.

Applications that are incompatible with network load balancing have one or more of the following characteristics:

- They bind to actual computer names (examples of such applications are Microsoft Exchange Server and Distributed File System).
- They have files that must be continuously open for writing (examples of such applications are Exchange Server and Simple Mail Transfer Protocol (SMTP) servers).

Note Before you load balance an application in a Network Load Balancing service cluster, review the application license or check with the application vendor. The application vendor can set licensing policies for applications that are running on clusters.

Services

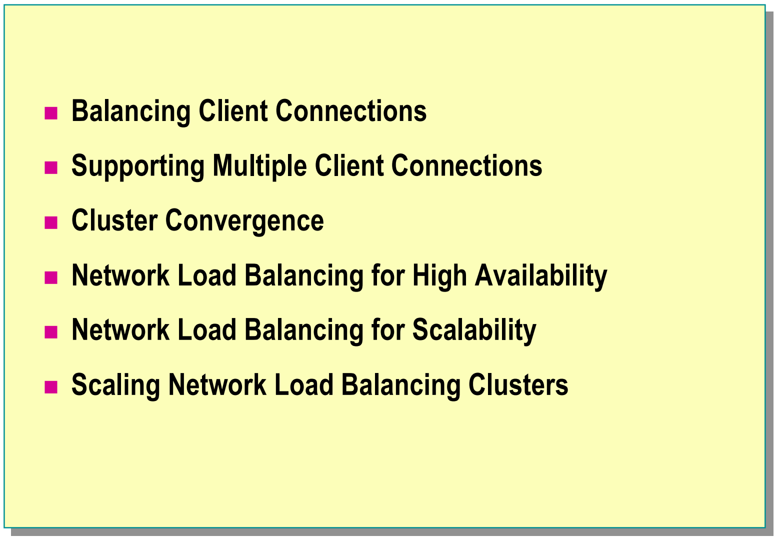
In addition to knowing what applications benefit from a clustering solution, there are services that have been identified as being compatible with Network Load Balancing. To modify the default behavior of these services, you can create port rules that cover specific port ranges. The following table below lists some examples of services and their associated ports.

Protocol	Port Number	Product Information
HTTP	Port 80	Hypertext Transfer Protocol Web servers, such as Microsoft Internet Information Services
HTTPS	Port 443	HTTP over Secure Sockets Layer (SSL) for encrypting Web traffic
FTP	Port 20, Port 21, Ports 1024-65535	File Transfer Protocol
TFTP	Port 69	Trivial File Transfer Protocol servers, which are used by applications such as the Bootstrap protocol (BOOTP)
SMTP	Port 25	Simple Mail Transport Protocol (SMTP), which is used by applications such as Microsoft Exchange Server
Microsoft Terminal Services	Port 3389	

◆ Network Load Balancing Functionality

Topic Objective
To describe Network Load Balancing functionality.

Lead-in
Using the functionality of the Network Load Balancing driver, you can load balance client connections by configuring the driver to manually or evenly balance the incoming IP traffic.



- **Balancing Client Connections**
- **Supporting Multiple Client Connections**
- **Cluster Convergence**
- **Network Load Balancing for High Availability**
- **Network Load Balancing for Scalability**
- **Scaling Network Load Balancing Clusters**

Using the functionality of the Network Load Balancing driver, you can configure the driver to distribute inbound client IP traffic across cluster members by using the following strategies:

- Evenly distributed
- Manually distributed
- Distribution based on host priority

The priority selection is also seen in a process known as cluster convergence, where a failed cluster host breaks the intercommunications between the hosts and the driver invokes a convergence algorithm. The IP traffic is then redistributed away from the failed host to the remaining hosts that are still active in the cluster.

Convergence results in high availability of the IP-based services, because the client connections are automatically redistributed within the cluster. Network Load Balancing is a high availability alternative to round robin Domain Name System (DNS), which will continue to route IP traffic to a failed host until it is manually removed from DNS.

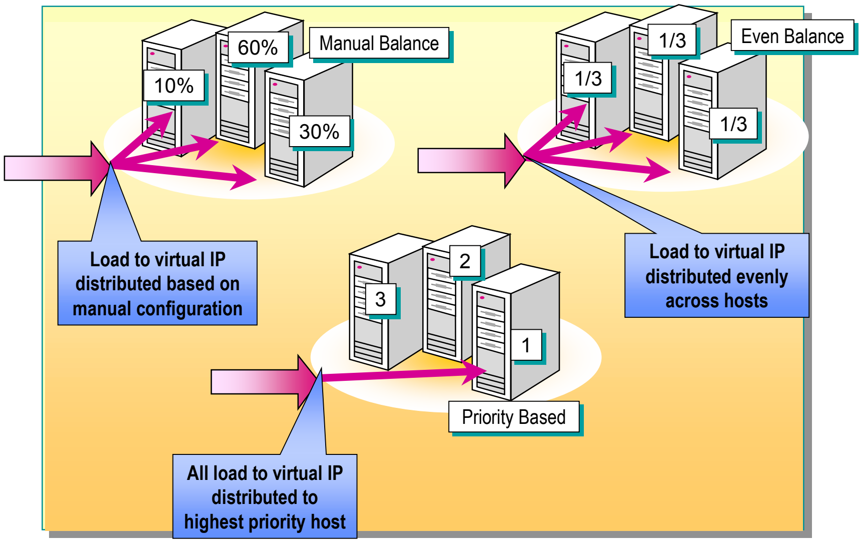
With Network Load Balancing you can manage multiple client connections and their session state. You are required to determine if your application instances can share client state to all of the hosts in the cluster. To resolve client state errors, which might occur with applications that cannot share state, you can configure the Network Load Balancing driver to handle all of the TCP client connections on the same cluster host.

When client connection requests exceed your system capacity, you can scale your Network Load Balancing cluster by adding hosts to meet performance requirements.

Balancing Client Connections

Topic Objective
To describe the functionality of the Network Load Balancing driver.

Lead-in
The Network Load Balancing driver manages client connections by allowing the system administrator to manually configure the load or distribute the load evenly across all of the hosts or to the highest priority host.


60%
Manual Balance
10%
30%
Load to virtual IP distributed based on manual configuration
1/3
Even Balance
1/3
1/3
Load to virtual IP distributed evenly across hosts
2
3
1
Priority Based
All load to virtual IP distributed to highest priority host

******************************ILLEGAL FOR NON-TRAINER USE******************************

The Network Load Balancing driver manages client connections by allowing you to manually configure the load or distribute the load evenly across all of the hosts or to the highest priority host. By combining the manageability and the functionality of the Network Load Balancing driver, you can assign a virtual IP address, implement the Network Load Balancing driver across all of the hosts in the cluster, and redistribute client traffic.

Manual Balance

The Network Load Balancing driver enables you to assign a virtual IP address to a group of (up to 32) hosts running the Network Load Balancing driver. This group of hosts, also known as a cluster, appears as a single system image to clients. Although Network Load Balancing requires only a single virtual IP address, it can support many virtual IP addresses for the cluster.

Even Balance

The fully distributed implementation of the Network Load Balancing driver allows it to run simultaneously on every host in the cluster. If all but one of the cluster hosts fails, the cluster will continue to provide service to connecting clients.

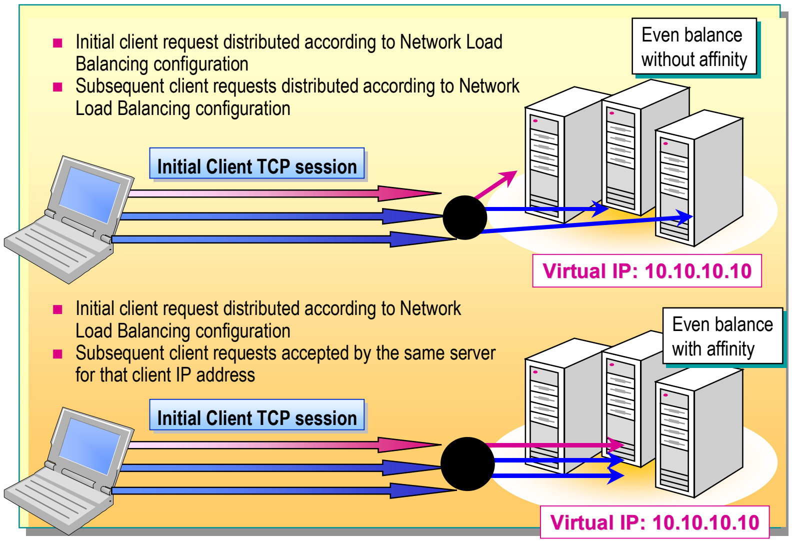
Priority Based

The Network Load Balancing driver automatically detects hosts that have become unavailable and redistributes traffic among surviving cluster hosts within eight seconds. The system administrator can establish the priority-based distribution during driver configuration. Each of the host members in the cluster will be given a specific priority number (1-32) by the administrator. During failover the Network Load Balancing driver will route the inbound IP traffic to the next host with the highest priority.

Supporting Multiple Client Connections

Topic Objective
To identify support issues for client connections in a Network Load Balancing cluster.

Lead-in
In a load-balanced, multiserver environment, managing and resolving client, application, and session state for individual clients can be complex.



- Initial client request distributed according to Network Load Balancing configuration
- Subsequent client requests distributed according to Network Load Balancing configuration

Even balance without affinity

Initial Client TCP session

Virtual IP: 10.10.10.10

- Initial client request distributed according to Network Load Balancing configuration
- Subsequent client requests accepted by the same server for that client IP address

Even balance with affinity

Initial Client TCP session

Virtual IP: 10.10.10.10

In a load-balanced multiserver environment, managing and resolving client, application, and session state for individual clients can be complex. By default, in a network load balancing solution, different hosts in the cluster can service multiple client connections.

When a client creates an initial connection to a host in the cluster, the application running on this host holds the client state. If the same host does not service subsequent connections from the client, errors can occur if the application instances do not share the client state between hosts.

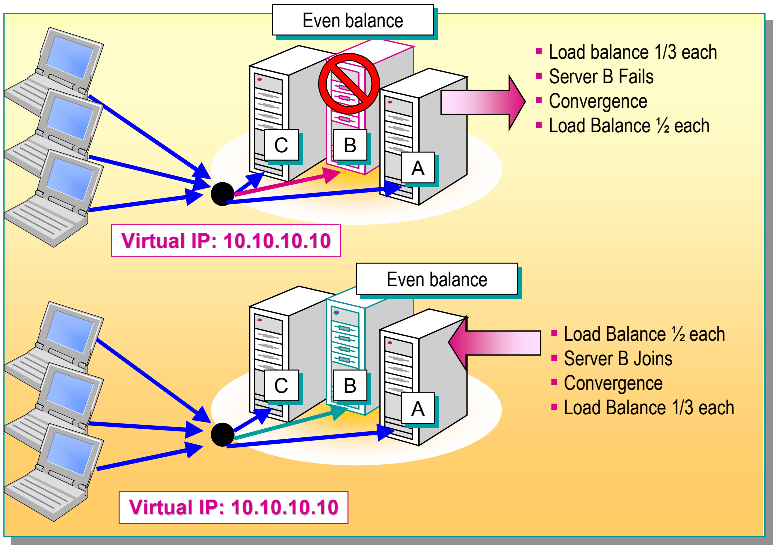
For example, application development for an ASP-based Web site can be more difficult if the application must share the client state among the multiple hosts in the cluster. If in the preceding graphic all of the client connections can be guaranteed to go to the same server, you can solve the difficulties with the application that is not sharing the client state among host instances.

Using a Network Load Balancing feature called affinity, you can ensure that the same cluster host handles all of the TCP connections from one client IP address. Affinity allows you to scale applications that manage session state spanning multiple client connections. In a Network Load Balancing cluster, with affinity selected, initial client connection requests are distributed according to the cluster configuration, but after you have established the initial client request the same host will service all of the subsequent requests from that client.

Cluster Convergence

Topic Objective
To describe interhost communications and convergence within a Network Load Balancing cluster.

Lead-in
When the state of the Network Load Balancing cluster changes (hosts fail, leave, or join the cluster) Network Load Balancing invokes convergence.



- Load balance 1/3 each
- Server B Fails
- Convergence
- Load Balance ½ each

- Load Balance ½ each
- Server B Joins
- Convergence
- Load Balance 1/3 each

Key Points
Convergence after you have added a new host may affect client sessions where client affinity is required because clients may be remapped to different cluster hosts between multiple connections.

When the state of the Network Load Balancing cluster changes (hosts fail, leave or join the cluster) Network Load Balancing invokes convergence.

The continuous interhost communication between cluster hosts, also known as heartbeat messages, invokes convergence and Network Load Balancing elects the host with the highest priority as the new default host.

During convergence, the hosts continue to handle incoming network traffic as usual, except that traffic for a failed host does not receive service. At the completion of convergence, client traffic for a failed host is redistributed to the remaining hosts.

If you add a host to the cluster, convergence allows this host to receive its share of load-balanced traffic. Expansion of the cluster does not affect ongoing cluster operations and is achieved in a manner transparent to both Internet clients and to server programs. If a host attempts to join the cluster with an incompatible configuration, completion of convergence is inhibited, and the host does not join the cluster. Thus an improperly configured host is prevented from handling cluster traffic.

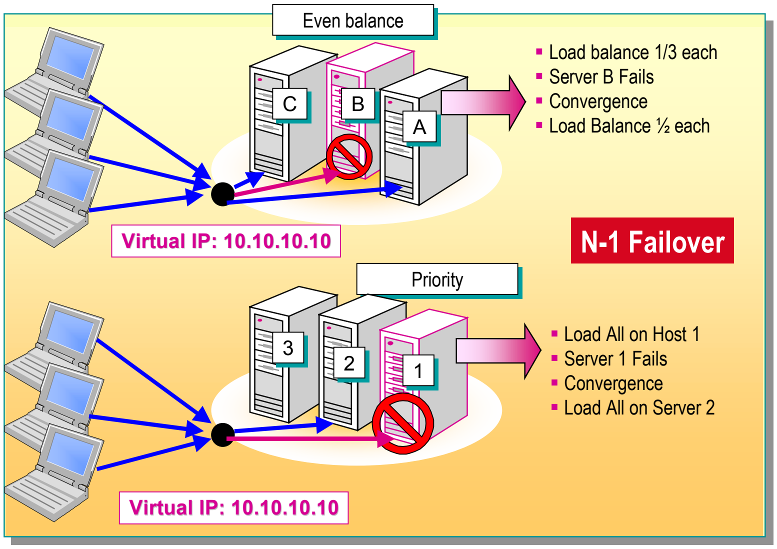
Note Convergence after you have added a new host may affect client sessions where client affinity is required because clients may be remapped to different cluster hosts between multiple connections.

When all of the cluster hosts have reached consensus on the correct new state of the cluster, they record the change in cluster membership in the Windows 2000 event log and begin to process traffic.

Network Load Balancing for High Availability

Topic Objective
To identify the high availability features of a Network Load Balancing cluster.

Lead-in
Network Load Balancing manages TCP/IP traffic to maintain high availability and dynamic load balancing for IP-based services.



- Load balance 1/3 each
- Server B Fails
- Convergence
- Load Balance ½ each

N-1 Failover

- Load All on Host 1
- Server 1 Fails
- Convergence
- Load All on Server 2

Delivery Tip
Use the graphic to explain the concept of high availability within a Network Load Balancing cluster. If you have an *n*-1 failover with one cluster that has IP traffic evenly distributed, the other cluster host is set to failover to the cluster host with the highest priority.

Network Load Balancing manages TCP/IP traffic to maintain high availability and dynamic load balancing for IP-based services. When a host fails or goes offline, Network Load Balancing automatically reconfigures the cluster to direct client requests to the remaining computers. In addition, for load-balanced ports, the load is automatically redistributed among the computers still operating, and ports with a single server have their traffic redirected to a specific host. Such redistribution of the workload typically takes less than ten seconds and is referred to as cluster convergence.

To maximize throughput and availability, Network Load Balancing uses fully distributed software architecture. This enhanced availability results from (*n*-1)-way failover in a cluster with *n* hosts. Maximizing throughput means that the Network Load Balancing functionality allows the cluster to dynamically respond to reconfiguration because of a host failure or an administrator adding or removing a host.

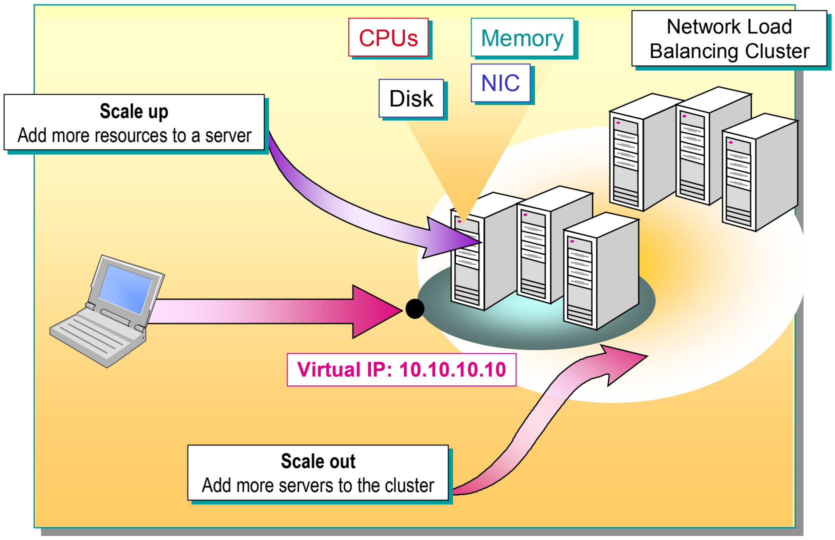
When a host failure occurs, connections to the failed or offline server are lost. When the client re-establishes these connections to the cluster, they will be distributed to members of the cluster who are currently online. After the necessary maintenance is completed, the offline computer can transparently rejoin the cluster and regain its share of the workload. This robust fault tolerance avoids the single points of failure or performance bottlenecks of other load balancing solutions. Network Load Balancing distributes the client connection load within the cluster by using the following strategies:

- Divides the load of incoming client requests evenly among the hosts.
- Specifies that one host handles all of the client requests with the others serving as failover alternatives.

Network Load Balancing for Scalability

Topic Objective
To identify the scaling methods for Network Load Balancing clusters.

Lead-in
Network Load Balancing provides scalability to enterprise-wide TCP/IP service such as Web, Terminal Services, proxy, VPN, and streaming media services.


CPUs
Memory
Network Load Balancing Cluster
NIC
Disk
Scale up
Add more resources to a server
Virtual IP: 10.10.10.10
Scale out
Add more servers to the cluster

****************************ILLEGAL FOR NON-TRAINER USE******************************

Delivery Tip
Use the graphic to explain the concept of scalability to the students. They can scale up by adding more resources to a cluster host or they can scale out by adding more servers to the cluster.

Network Load Balancing provides scalability to enterprise wide TCP/IP services such as Web, Terminal Services, proxy, Virtual Private Network (VPN), and streaming media services. Network Load Balancing cluster hosts intercommunicate to provide one of the key benefits, scalability.

Network Load Balancing scales the performance of a server-based program, such as a Web server, by distributing its client requests across multiple identical servers within the cluster; you can add more servers to the cluster as traffic increases. Up to 32 servers are possible in any one cluster.

You can improve the performance of each individual host in a cluster by adding more or faster CPUs, network adapters and disks, and in some cases by adding more memory. These additions to the Network Load Balancing cluster is termed scaling up, but requires more intervention and careful planning than scaling out. Limitations of applications or the operating system configuration could mean that scaling up by adding more memory may not be as appropriate as scaling out.

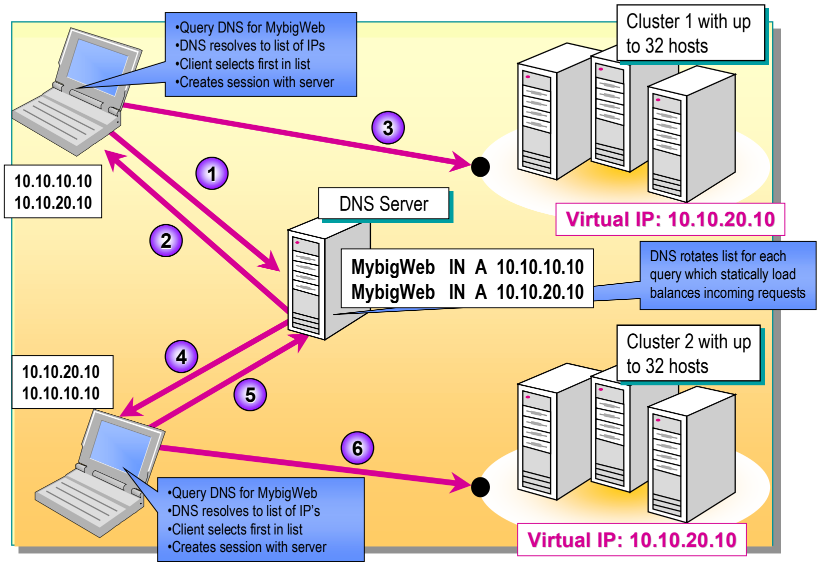
You can handle additional IP traffic by simply adding computers to the Network Load Balancing cluster as necessary. Load balancing, in conjunction with the use of server farms, is part of a scaling approach referred to as scaling out. The greater the number of computers involved in the load-balancing scenario, the higher the throughput of the overall server farm.

Tip On a system where kernel resources, such as page table entries, non-paged pool, and paged pool, are limited and tuning is not effective, it is more appropriate to scale out than to scale up.

Scaling Network Load Balancing Clusters

Topic Objective
To describe load balancing cluster scaling.

Lead-in
Network Load Balancing clusters have a maximum of 32 hosts, and all of the hosts must be on the same subnet.


•Query DNS for MybigWeb
•DNS resolves to list of IPs
•Client selects first in list
•Creates session with server
Cluster 1 with up to 32 hosts
3
1
10.10.10.10
10.10.20.10
DNS Server
Virtual IP: 10.10.20.10
2
MybigWeb IN A 10.10.10.10
MybigWeb IN A 10.10.20.10
DNS rotates list for each query which statically load balances incoming requests
Cluster 2 with up to 32 hosts
4
10.10.20.10
10.10.10.10
5
6
•Query DNS for MybigWeb
•DNS resolves to list of IP's
•Client selects first in 10
•Creates session with server
Virtual IP: 10.10.20.10

Delivery Tip
Review the graphic with the students (use the text explanation) to describe how round robin DNS can be combined with Network Load Balancing to provide scalability and high availability.

Network Load Balancing clusters have a maximum of 32 hosts, and all of the hosts must be on the same subnet. If a cluster cannot meet the performance requirements of a clustered application, such as a Web site, because of a host count or subnet throughput limitation, then you can use multiple clusters to scale out further.

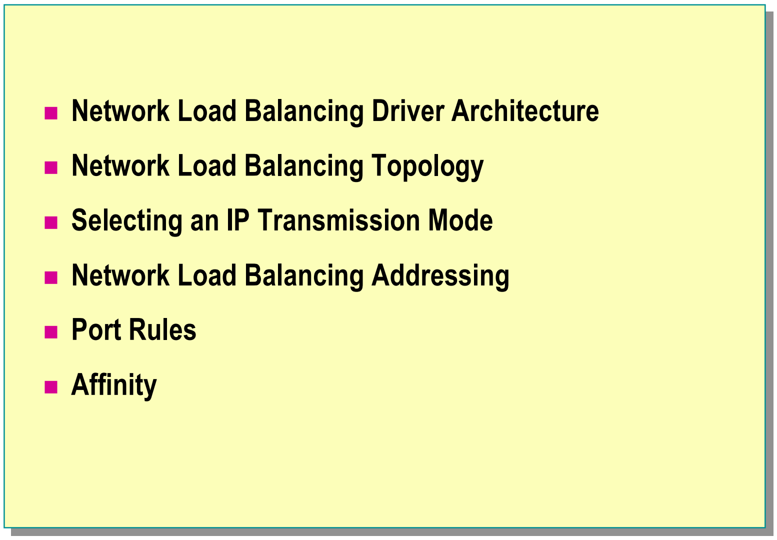
Combining round robin DNS and Network Load Balancing results in a very scalable and highly available configuration. Configuring multiple Network Load Balancing clusters on different subnets and configuring DNS to sequentially distribute requests across multiple Network Load Balancing clusters can evenly distribute the client load that is distributed across several clusters. When multiple Network Load Balancing Web clusters are configured with round robin DNS, the Web servers are made resilient to networking infrastructure failures also. For example, as shown in the graphic where arrows 1, 2, 3 and 4, 5, 6 represent a client URL query, DNS response, and cluster connection request, notice that each DNS entry is a reliable cluster and loss of an individual cluster member does not result in clients being issued nonfunctional IP addresses.

When you use round robin DNS in conjunction with Network Load Balancing clusters, each cluster is identified in DNS by the cluster virtual IP. Because each cluster is automatically capable of both load balancing and fault tolerance, each DNS-issued IP address will function until all hosts in that particular cluster fail. Round robin DNS enables only a limited form of TCP/IP load balancing for IP-based servers when used without Network Load Balancing. When used with multiple individual hosts, such as Web servers, round robin DNS does not function effectively as a high-availability solution. If a host fails, round robin DNS continues to route requests to the failed server until the server is removed from DNS.

◆ Network Load Balancing Architecture

Topic Objective
To identify the components of Network Load Balancing.

Lead-in
You will need to consider several architectural and configuration components of a network load balancing solution when installing the Network Load Balancing driver.



- **Network Load Balancing Driver Architecture**
- **Network Load Balancing Topology**
- **Selecting an IP Transmission Mode**
- **Network Load Balancing Addressing**
- **Port Rules**
- **Affinity**

****************************ILLEGAL FOR NON-TRAINER USE******************************

You will need to consider several architectural and configuration components of a network load balancing solution when installing the Network Load Balancing driver.

Network Load Balancing is automatically installed and can be optionally enabled on the Advanced Server and Datacenter Server versions of the Windows 2000 operating system. It operates as an optional service for local area network (LAN) connections and can be enabled for one LAN connection in the system; this LAN connection is known as the cluster adapter.

Network Load Balancing does not require any hardware changes to install and run it. Because it is compatible with almost all Ethernet and Fiber Distributed Data Interface (FDDI) network adapters, it has no specific hardware compatibility list (HCL).

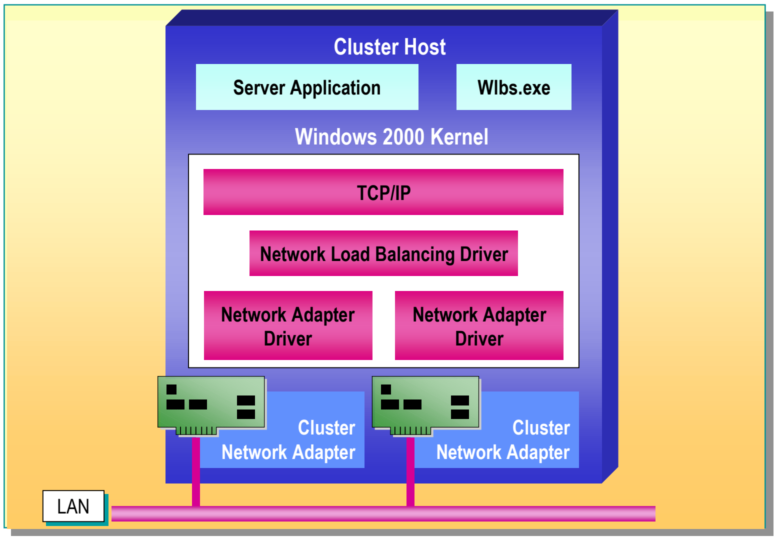
It is important that you understand each of the following components to implement an effective network load balancing solution. These components include:

- *Network Load Balancing driver architecture*. Focuses on the Network Load Balancing driver, its configuration in relation to the TCP/IP stack, and how it operates.
- *Topology*. Addresses the network type and configuration, and how the client requests are routed to cluster hosts.
- *Addressing*. Focuses on the selection of the virtual and dedicated IP addresses that are used for a host.
- *Network Load Balancing Parameters.* Focuses on the driver configuration parameters to control client connections. These driver configuration parameters include port rules and affinity.

Network Load Balancing Driver Architecture

Topic Objective
To describe the Network Load Balancing driver architecture.

Lead-in
Network Load Balancing runs as a network driver logically situated beneath higher-level application protocols, such as HTTP and FTP.



Key Points
On each cluster host, the Network Load Balancing driver acts as a filter between the cluster adapter driver and the TCP/IP stack to allow the cluster host to receive a portion of the incoming network traffic.

Network Load Balancing runs as a network driver logically situated beneath higher-level application protocols, such as HTTP and FTP. On each cluster host, the driver acts as a filter between the network adapter's driver and the TCP/IP stack; the Network Load Balancing driver partitions and load balances incoming client requests among the cluster hosts.

To maximize throughput and availability, Network Load Balancing uses fully distributed software architecture, and an identical copy of the Network Load Balancing driver that runs in parallel on each host in a cluster.

Network Load Balancing Driver

You install the Network Load Balancing driver on a Windows 2000-based computer that is configured with TCP/IP, and is bound to a single network interface called the cluster adapter.

You configure the driver with a single IP address, the cluster primary IP address, on a single subnet for all of the cluster hosts. Each host has an identical Media Access Control (MAC) address that allows the hosts to concurrently receive incoming network traffic for the cluster's primary IP address (and for additional IP addresses on multihomed hosts).

Rule-Based Filter

The Network Load Balancing driver partitions and load balances incoming client requests among the cluster hosts, acting as a rule-based filter between the network adapter's driver and the TCP/IP stack. Each host receives a designated portion of the incoming network traffic.

System Resource Requirements

Windows 2000 Network Load Balancing operates on FDDI or Ethernet-based local area networks within the cluster. It has been tested on 10 megabits per second (Mbps), 100 Mbps, and gigabit Ethernet networks with any HCL-approved network adapters.

Network Load Balancing uses between 250 kilobytes (KB) and 4 megabytes (MB) of random access memory (RAM) during operation, based on the default parameters and the network load. You can modify the registry parameters to allow up to 15 MB of RAM to be used.

Distributed Architecture

Network Load Balancing is a distributed architecture, with an instance of the driver installed on each cluster host. Throughput is maximized to all cluster hosts by eliminating the need to route incoming packets to individual cluster hosts, through a process called filtering. Filtering out unwanted packets in each host improves throughput; this process is faster than routing packets (which involves receiving, examining, rewriting, and resending the packets).

Another key advantage to the Network Load Balancing fully distributed architecture is the enhanced availability resulting from (*n*-1) way failover in a cluster with *n* hosts. In contrast, dispatcher-based solutions create an inherent single point of failure that you must eliminate by using a redundant dispatcher that provides only one-way failover. Dispatcher-based solutions offer a less robust failover solution than does a fully distributed architecture.

Connection Management Algorithm

The Network Load Balancing driver uses a fully distributed filtering algorithm to statistically map incoming client connections to the cluster hosts, based upon their IP address, port, and other information.

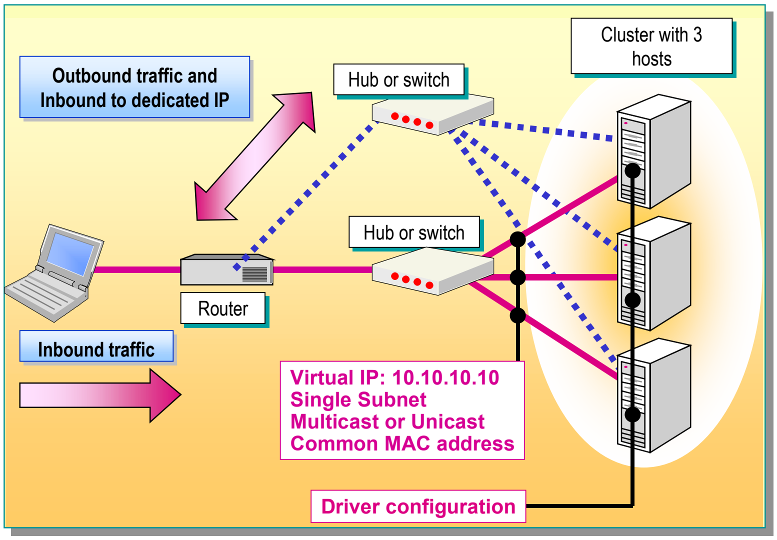
When receiving an incoming packet, all hosts within the cluster simultaneously perform this mapping to determine which host should handle the packet. Those hosts not required to service the packet simply discard it. The mapping remains constant unless the number of cluster hosts changes or the filter processing rules change.

The filtering algorithm is much more efficient in its packet handling than centralized load balancers, which must modify and retransmit packets. Efficient packet handling allows for a much higher aggregate bandwidth to be achieved on industry standard hardware.

Network Load Balancing Topology

Topic Objective
To describe cluster configuration.

Lead-in
Considering network configuration and required cluster functionality when implementing a cluster enables you to choose the most appropriate cluster configuration.



Delivery Tip
Use the graphic to explain the topology of the Network Load Balancing driver.

Considering network configuration and required cluster functionality when implementing a cluster enables you to choose the most appropriate cluster configuration. Network Load Balancing requires at least one network adapter; and different hosts in a cluster can have a different number of adapters, but all must use the same network IP transmission mode, either unicast or multicast.

Unicast and Multicast Modes

You can configure the Network Load Balancing driver to operate in one of two modes: unicast and multicast. Unicast support is the default setting.

When you enable unicast support, the unicast mode changes the cluster adapter's MAC address to the cluster MAC address. This cluster address is the same MAC address that is used on all cluster hosts. When this change is made, clients can no longer address the cluster adapters by their original MAC addresses.

When you enable multicast support, Network Load Balancing adds a multicast MAC access to the cluster adapters on all of the cluster hosts. At the same time, the cluster adapters retain their original MAC addresses.

Note You cannot change the MAC addresses on some network adapters; check the hardware specifications for your network adapter.

If clients are accessing a Network Load Balancing cluster through a router when the cluster has been configured to operate in multicast mode, be sure that the router meets the following requirements:

- Accepts an Address Resolution Protocol (ARP) reply that has one MAC address in the payload of the ARP structure but appears to arrive from a station with another MAC address, as judged by the Ethernet header.
- Accepts an ARP reply that has a multicast MAC address in the payload of the ARP structure.

If your router does not meet these requirements, you can create a static ARP entry in the router. For example, Cisco routers require a static ARP entry, because they do not support the resolution of unicast IP addresses to multicast MAC addresses.

Note The Network Load Balancing driver does not support a mixed unicast/multicast environment. All cluster hosts must be either multicast or unicast; otherwise, the cluster will not function properly.

Subnet and Network Considerations

The Network Load Balancing architecture with a single MAC address for all cluster hosts maximizes use of the subnet's hub and/or switch architecture to simultaneously deliver incoming network traffic to all cluster hosts.

Your network configuration will typically include routers, but may also include layer 2 switches (collapsed backbone) rather than the simpler hubs or repeaters that are available. Cluster configuration, when using hubs, is predictable because the hubs distribute IP traffic to all ports.

Note If client-side network connections at the switch are significantly faster than server-side connections, incoming traffic can occupy a prohibitively large portion of server-side port bandwidth.

Network Adapters

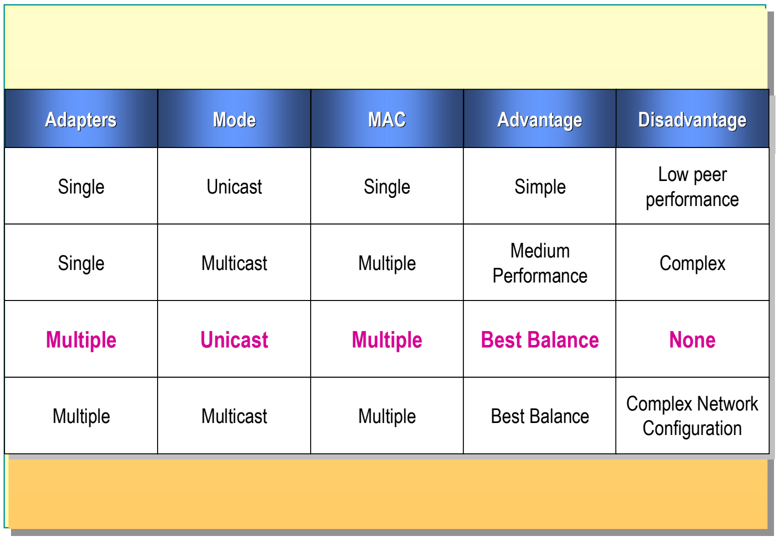
Network Load Balancing requires only a single network adapter, but for optimum cluster performance, you should install a second network adapter on each Network Load Balancing host. In this configuration, one network adapter handles the network traffic that is addressed to the server as part of the cluster. The other network adapter carries all of the network traffic that is destined to the server as an individual computer on the network, including cluster interhost communications.

Note Network Load Balancing with a single network adapter can provide full functionality if you enable multicast support for this adapter.

Selecting an IP Transmission Mode

Topic Objective
To describe the unicast and multicast environments.

Lead-in
When you are implementing a Network Load Balancing cluster, the Internet Protocol transmission mode selected and the number of network adapters required will depend on network requirements.



Adapters	Mode	MAC	Advantage	Disadvantage
Single	Unicast	Single	Simple	Low peer performance
Single	Multicast	Multiple	Medium Performance	Complex
Multiple	**Unicast**	**Multiple**	**Best Balance**	**None**
Multiple	Multicast	Multiple	Best Balance	Complex Network Configuration

****************************ILLEGAL FOR NON-TRAINER USE******************************

When you are implementing a Network Load Balancing solution, the Internet Protocol transmission mode that is selected and the number of network adapters that are required are dependent upon the following network requirements:

Delivery Tip
Review the advantages and disadvantages of the each IP transmission mode using the graphic and the text explanation for each mode.

- Layer 2 switches or hubs
- Interhost peer-to-peer communications
- Maximized communication performance

For example, a cluster supporting a static Hypertext Markup Language (HTML) Web application can have a requirement to synchronize the Web site copies of a large number of cluster hosts. This scenario requires interhost peer-to-peer communications. You select the number of network adapters and the IP communications mode to meet this requirement.

There is no restriction on the number of network adapters, and different hosts can have a different number of adapters. You can configure Network Load Balancing to use one of four different models.

Single Network Adapter in Unicast Mode

The single network adapter in unicast mode is suitable for a cluster in which you do not require ordinary network communication among cluster hosts, and in which there is limited dedicated traffic from outside the cluster subnet to specific cluster hosts. In this model, the computer can also handle traffic from inside the subnet if the IP datagrams do not carry the same MAC address as on the cluster adapter.

Single Network Adapter in Multicast Mode

This model is suitable for a cluster in which ordinary network communication among cluster hosts is necessary or desirable, but in which there is limited dedicated traffic from outside the cluster subnet to specific cluster hosts.

Multiple Network Adapter in Unicast Mode

This model is suitable for a cluster in which ordinary network communication among cluster hosts is necessary or desirable, and in which there is comparatively heavy dedicated traffic from outside the cluster subnet to specific cluster hosts.

Important The Multiple Network Adapter in Unicast Mode is the preferred configuration used by most sites, because a second network adapter may enhance overall network performance.

Multiple Network Adapter in Multicast Mode

This model is suitable for a cluster in which ordinary network communication among cluster hosts is necessary, and in which there is heavy dedicated traffic from outside the cluster subnet to specific cluster hosts.

Comparison of Modes

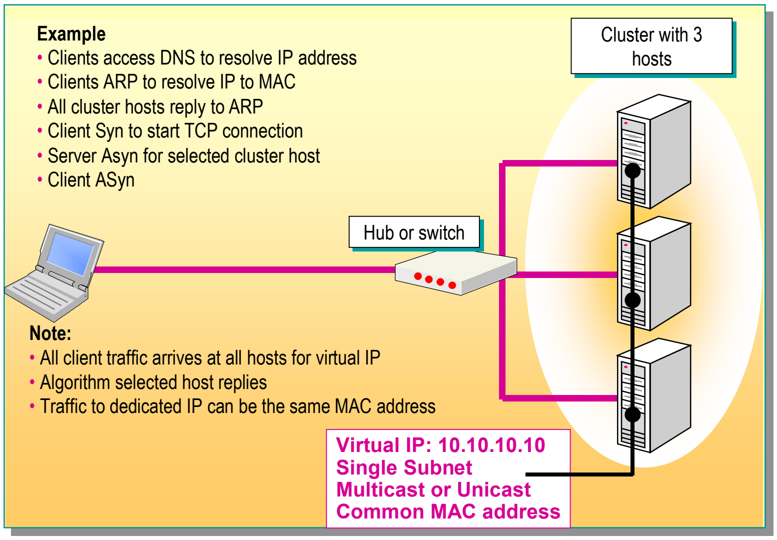
The advantages and disadvantages of each model are listed in the table below.

Adapters	**Mode**	**Advantages (+)/Disadvantages (-)**
Single	Unicast	+ Simple configuration + Limited routed peer-to-peer communications - No peer-to-peer cluster host communications - Poor overall performance
Single	Multicast	+ Medium routed peer-to-peer communications - Complex network considerations/configuration - Medium overall performance
Multiple	*Unicast*	+ *Recommended configuration* + High routed peer-to-peer communications + Works with all routers + High performance
Multiple	Multicast	+ High performance + High routed peer-to-peer communications - Complex network considerations/configuration

Network Load Balancing Addressing

Topic Objective
To identify the features of the primary and dedicated IP addresses.

Lead-in
After you have enabled Network Load Balancing, you configure its parameters by using the **Properties** dialog box.



Example
- Clients access DNS to resolve IP address
- Clients ARP to resolve IP to MAC
- All cluster hosts reply to ARP
- Client Syn to start TCP connection
- Server Asyn for selected cluster host
- Client ASyn

Note:
- All client traffic arrives at all hosts for virtual IP
- Algorithm selected host replies
- Traffic to dedicated IP can be the same MAC address

Virtual IP: 10.10.10.10
Single Subnet
Multicast or Unicast
Common MAC address

****************************ILLEGAL FOR NON-TRAINER USE******************************

Delivery Tip
Ensure that the students understand all of the concepts presented. This page describes in detail the functionality of the Network Load Balancing driver.

After you have enabled Network Load Balancing, you configure its parameters by using the **Properties** dialog box. The Network Load Balancing cluster is assigned a primary Internet Protocol address. This IP address represents a virtual IP address to which all of the cluster hosts respond and the remote control program that is provided with Network Load Balancing uses this IP address to identify a target cluster.

Primary IP Address

The primary IP address is the virtual IP address of the cluster and must be set identically for all hosts in the cluster. You can use the virtual IP address to address the cluster as a whole. The virtual IP address is also associated with the Internet name that you specify for the cluster.

Dedicated IP Address

You can also assign each cluster host a dedicated IP address for network traffic that is designated for that particular host only. Network Load Balancing never load-balances the traffic for the dedicated IP addresses, it only load-balances incoming traffic from all IP addresses other than the dedicated IP address.

When you configure the Network Load Balancing driver, it is important to enter the dedicated IP address, the primary IP address, and other optional virtual IP addresses into the TCP/IP **Properties** dialog box. Entering the virtual IP addresses into the **Properties** dialog box will enable the host's TCP/IP stack to respond to these IP addresses.

Distribution of Traffic Within the Cluster

When the virtual IP address is resolved to the station address (MAC address), this MAC address is common for all hosts in the cluster. You can enable client connections to only the required cluster host when more packets are sent. The responding host then substitutes a different MAC address for the inbound MAC address in the reply traffic. The substitute MAC address is referred to as the Source MAC address. The table shows the MAC addresses that will be generated for a cluster adapter.

IP Mode	MAC Address	Explanation
Unicast inbound	02-BF-W-X-Y-Z	W-X-Y-Z = IP address Onboard MAC disabled
Multicast inbound	03-BF-W-X-Y-Z	W-X-Y-Z = IP Address Onboard MAC enabled

Source outbound	02-P-W-X-Y-Z	W-X-Y-Z = IP Address
		P = Host Priority

In the unicast mode of operation, the Network Load Balancing driver disables the onboard MAC address for the cluster adapter. You cannot use the dedicated IP address for interhost communications, because all of the hosts have the same MAC address.

In multicast mode of operation, the Network Load Balancing driver supports both the onboard and the multicast address. If your cluster configuration will require connections from one cluster host to another, for example, when making a NetBIOS connection to copy files, use multicast mode or install a second network interface card (NIC).

If the cluster hosts were attached to a switch instead of a hub, the use of a common MAC address would create a conflict because layer-2 switches expect to see unique source MAC addresses on all switch ports. To avoid this problem, Network Load Balancing uniquely modifies the source MAC address for outgoing packets; a cluster MAC address of 02-BF-1-2-3-4 is set to 02-*p*-1-2-3-4, where *p* is the host's priority within the cluster.

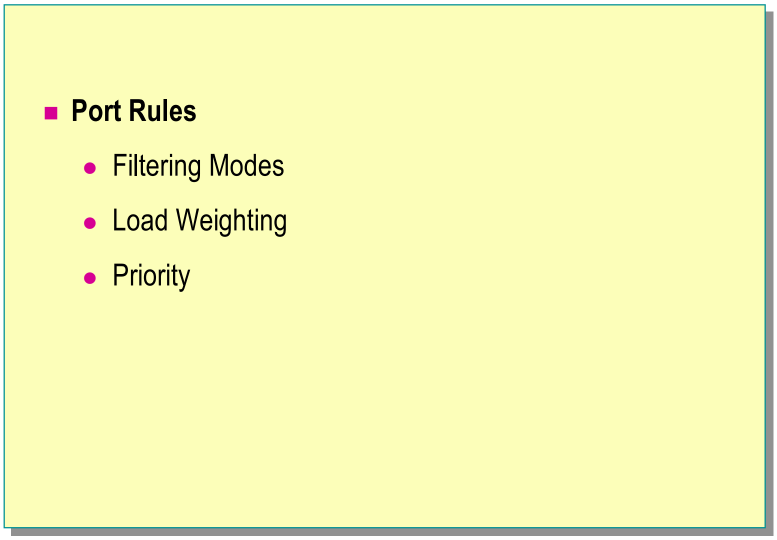
This technique prevents the switch from learning the cluster's inbound MAC address, and as a result, incoming packets for the cluster are delivered to all of the switch ports. If the cluster hosts are connected to a hub instead of to a switch, you can disable Network Load Balancing's masking of the source MAC address in unicast mode to avoid flooding upstream switches. You disable Network Load Balancing by setting the Network Load Balancing registry parameter *MaskSourceMAC* to 0. The use of an upstream level three switch will also limit switch flooding.

The unicast mode of Network Load Balancing induces switch flooding to simultaneously deliver incoming network traffic to all of the cluster hosts. Also, when Network Load Balancing uses multicast mode, switches often flood all of the ports by default to deliver multicast traffic. However, the multicast mode of Network Load Balancing gives the system administrator the opportunity to limit switch flooding by configuring a virtual LAN within the switch for the ports corresponding to the cluster hosts.

Port Rules

Topic Objective
To describe the port rule parameters.

Lead-in
You will create port rules for individual ports and for groups of ports that are required for particular applications and services.



- **Port Rules**
  - Filtering Modes
  - Load Weighting
  - Priority

You will create port rules for individual ports and for groups of ports that Network Load Balancing requires for particular applications and services. The filter setting then defines whether the Network Load Balancing driver will pass or block the traffic.

Key Points
You can add port rules or update parameters by taking each host out of the cluster in turn, updating its parameters, and then returning it to the cluster.
The host joining the cluster handles no traffic until convergence is complete.
The cluster does not converge to a consistent state until all of the hosts have the same number of rules.
For example, if a rule is added, it does not take effect until you have updated all of the hosts and they have rejoined the cluster.

The Network Load Balancing driver controls the distribution and partitioning of TCP and UDP traffic from connecting clients to selected hosts within a cluster by passing or blocking the incoming data stream for each host. Network Load Balancing does not control any incoming IP traffic other than TCP and UDP for ports that a port rule specifies.

You can add port rules or update parameters by taking each host out of the cluster in turn, updating its parameters, and then returning it to the cluster. The host joining the cluster handles no traffic until convergence is complete. The cluster does not converge to a consistent state until all of the hosts have the same number of rules. For example, if a rule is added, it does not take effect until you have updated all of the hosts have been updated and they have rejoined the cluster.

Note Internet Control Message Protocol (ICMP), Internet Group Membership Protocol (IGMP), ARP, or other IP protocols are passed unchanged to the TCP/IP protocol software on all of the hosts within the cluster.

Port rules define individual ports or groups of ports for which the driver has a defined action. You need to consider certain parameters when creating the port rules, such as the:

- TCP or UDP port range for which you should apply this rule.
- Protocols for which this rule should apply (TCP, UDP, or both).
- Filtering mode chosen: multiple hosts, single host, or disabled.

When defining the port rules, it is important that the rules be exactly the same for each host in the cluster because if a host attempts to join the cluster with a different number of rules from the other hosts, the cluster will not converge. The rules that you enter for each host in the cluster must have matching port ranges, protocol types, and filtering modes.

Filtering Modes

The filter defines for each port rule whether the incoming traffic is discarded, handled by only one host, or distributed across multiple hosts. The three possible filtering modes that you can apply to a port rule are:

- *Multiple hosts*. Specifies that multiple hosts in the cluster will handle network traffic for the associated port rule. You can specify that the cluster equally distribute the load among the hosts or that each host handle a specified load weight.
- *Single host*. Specifies that a single host handle the network traffic for the associated rule. This filtering mode provides fault tolerance for the handling of network traffic with the target host defined by its priority.
- *Disabled*. Specifies that all network traffic for the associated port rule be locked. This filtering mode lets you build a firewall against unwanted network access to a specific range of ports; the driver discards the unwanted packets.

Load Weighting

When the filter mode for a port rule is set to Multiple, the *Load Weight* parameter specifies the percentage of load-balanced network traffic that this host should handle for the associated rule. Allowed value ranges are from 0 to 100.

Note To prevent a host from handling any network traffic for a port rule, set the load weight to zero.

Because hosts can dynamically enter or leave the cluster, the sum of the load weights for all cluster hosts does not have to equal 100. The percentage of host traffic is computed as the local load percentage value divided by the load weight sum across the cluster.

If you balance the load evenly across all of the hosts with this port rule, you can specify an equal load distribution parameter instead of specifying a load weight parameter.

Priority

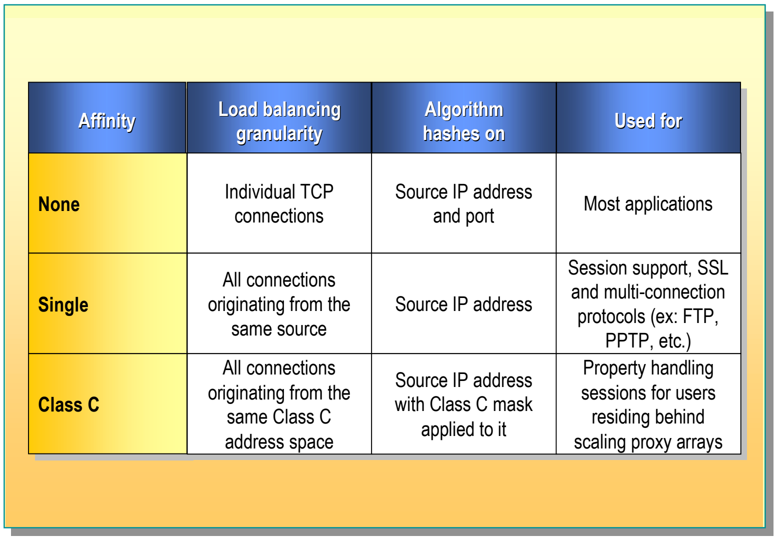
When the filter mode for a port rule is set to single, the priority parameter specifies the local host's network traffic for the associated port rule. The host with the highest handling priority for this rule among the current cluster members will handle all of the traffic.

The allowed values range from one, the highest priority, to the maximum number of hosts allowed, 32. This value must be unique for all hosts in the cluster.

Affinity

Topic Objective
To introduce the concepts of affinity.

Lead-in
Clients can have many TCP connections to a Network Load Balancing cluster; the load-balancing algorithm will potentially distribute these connections across multiple hosts in the cluster.



Affinity	Load balancing granularity	Algorithm hashes on	Used for
None	Individual TCP connections	Source IP address and port	Most applications
Single	All connections originating from the same source	Source IP address	Session support, SSL and multi-connection protocols (ex: FTP, PPTP, etc.)
Class C	All connections originating from the same Class C address space	Source IP address with Class C mask applied to it	Property handling sessions for users residing behind scaling proxy arrays

Clients can have many TCP connections to a Network Load Balancing cluster; the load-balancing algorithm will potentially distribute these connections across multiple hosts in the cluster.

If server applications have client or connection state information, this state information must be made available on all of the cluster hosts to prevent errors. If you cannot make state information available on all of the cluster hosts, you cannot use client affinity to direct all of the TCP connections from one client IP address to the same cluster host. Directing TCP connections from the IP address to the same host allows an application to maintain state information in the host memory.

For example: if a server application (such as a Web server) maintains state information about a client's site navigation status that spans multiple TCP connections, it is critical that all of the TCP connections for this client state information be directed to the same cluster host to prevent errors.

You can distribute incoming client connections based on the algorithm as determined by the following client affinity settings:

- *No Affinity*. Load distribution on a cluster is based on a distributed filtering algorithm that maps incoming client requests evenly across all of the cluster hosts.
- *Single Affinity*. All connection requests from a single IP client address will be directed to the same cluster host.
- *Class C Affinity*. The mapping algorithm bases load distributions on the Class C portion of the client's IP address.

Lab A: Planning an Installation

Topic Objective
To introduce the lab.

Lead-in
In this lab, you will use the planning worksheet guidelines to complete the planning worksheet.

Objectives

The purpose of this lab is to have you practice the planning steps for configuring a Network Load Balancing cluster.

After completing this lab, you will be able to:

- Select the appropriate applications and services.
- Determine the physical network constraints.
- Configure the physical components of the cluster.
- Configure the cluster for IP traffic.

Prerequisites

Before working on this lab, you must have:

- Completed Module 8, “Concepts of a Network Load Balancing Cluster.”

Estimated time to complete this lab: 45 minutes

Exercise 1
Planning Installation Worksheet

In this exercise, you will use the content you have learned from Module 8 and the Planning Installation Worksheet to complete this exercise.

Scenario

You are configuring a Network Load Balancing cluster to handle the network traffic for your organization’s planned Web site. The Web site will serve HTML Web pages with a mixture of graphics and text, but because the customers use e-mail to place online orders, no client state is maintained. The connection to the Internet is maintained separately from the connection that internal employees use to access the Internet. Use the following information to complete the planning worksheet:

- The clients accessing the Web site are expected to use approximately 8 megabits per second (Mbps) of bandwidth at peak times.
- The Internet connection is terminated in a firewall/proxy server array (three servers in the proxy array), and the internal connections from the proxy servers to the Web site will use 10 Mbps Ethernet connections.
- The internal network is a nonswitched environment using Simple Network Management Protocol (SNMP) managed hubs.
- A single subnet has been reserved for the Web site by using 10.10.20.10 (mask 255.255.255.0) as the cluster IP address. Selections for the dedicated IP address must be made from 10.10.20.50 and higher on the same subnet.
- There will be a staging server that is used to copy new Web pages to the production cluster members, by using a subnet on the private network. 10.10.25.30-10.10.25.45. (mask 255.255.255.0) that has been allocated for this purpose.
- To protect the Web servers as much as possible from external attack, you must restrict the inbound port availability as much as possible.
- The Web site must provide the highest possible availability and must be able to tolerate at least two servers in the cluster failing without interrupting service or reducing performance. (If the cluster must maintain performance when two servers are failed, then: 8/1.8 = 4.4, round up to 5 servers will be required when two are failed. Therefore the cluster must consist of seven servers total.)
- You have tested the Windows 2000-based Web servers and each server is expected to be able to handle throughput of 1.8 Mbps at the required performance level.

Completing the Planning Worksheet Guidelines

You will complete the Planning Worksheet for an installation of a Network Load Balancing cluster. If there are choices to make, they are listed for you and you can select them by circling them. For the other items, you will need to write in the required information.

► **To complete Part I Application or Service Information**

1. Decide on the applications and services that you wish the Network Load Balancing cluster to handle. Ensure that if you want to handle multiple applications and services, they are configurable on the same computer.
2. Determine if you have compatible applications and services for a Network Load Balancing cluster that use TCP connections or UDP data streams. You must determine if you are using TCP, UDP, or both.
3. Identify the inbound ports that your applications and services use. You use this information to define the traffic handling rules for the Network Load Balancing cluster.
4. Identify the outbound port usage. The outbound port usage is for reference only, but can be important in understanding data flows through firewalls and proxies.
5. Decide the purpose of your Network Load Balancing cluster. Is the purpose to increase system performance, to increase system capacity (scaling) or to achieve fault tolerance for the applications and services? This information will influence the configuration of the cluster and the traffic handling rules.

► **To complete Part II Physical Network Constraints**

1. After completing a risk audit for your network, applications and data you will have identified the single points of failure within your system. You must now determine which applications and services will benefit by being moved to the Network Load Balancing cluster.
2. Determine the aggregate throughput (capacity) requirements for your Network Load Balancing cluster based on the anticipated or projected client loads. Calculate the required throughput for the cluster as a whole. The network must be able to support this throughput requirement to the virtual IP for the cluster. You must consider all of the traffic that is using the routed path. If client throughput requirements are high (for example when supporting media or VPN servers) it is recommended that an isolated Internet connection support the traffic.
3. If after completing #2 you determine that the aggregate throughput (capacity) of the Network Load Balancing cluster exceeds the segment capacity you will need to select multiple clusters. The segment supporting the cluster members must provide bandwidth to meet the client requirements. If the segment will not support the aggregate bandwidth requirements use multiple clusters that are installed on separate segments to meet the requirements. Multiple clusters will require multiple A records in the DNS server.

4. Determine the throughput for each Network Load Balancing cluster server. You must calculate or empirically test the capability of each member of the cluster. You may rate each cluster host as supporting a particular client count or supporting a particular throughput rate. For example a VPN server can be rated to support 512 clients or can be rated at 3 Mbps. Providing a measurement of the load capability allows you to make decisions on load distribution within cluster members.
5. After identifying risks, the aggregate throughput for your system and for each server, you will need to determine how many hosts you will require in the Network Load Balancing cluster. You can calculate the number of hosts in a cluster based on the client throughput requirements or number of clients and the capability of each member.
6. If you are using Network Load Balancing in a priority-based failover mode rather than a fault tolerant load-balanced mode, each host must support all of the client connections. Where load is balanced across many members, each host must be able to support the designed client load for each member in addition to the load that would be distributed if a member host failed.
7. For example, in a cluster of ten members where each host is designed to support 100 client connections, the failure of one host would result in surviving members supporting an average of 112 client connections.

Note In this calculation the average number of connections was rounded to 112.

8. Determine if connections to the Internet and an intranet are required. If connections to the Internet, or any other public network are supported, you may have to consider security and bandwidth issues. If you are using only intranet connections, then you must consider security to the internal network design levels and bandwidth.
9. Determine if you have staging servers for your Web site and require interhost communications. If there is communication between cluster members or staging servers for synchronization, you must consider the impact this data flow will have on the virtual IP segment for the cluster. It is recommended that the segment supporting the virtual IP address carry only inbound and outbound cluster client traffic. Consider adding multiple network cards and separating the noncluster-related traffic on a separate subnet.
10. There is a minimum requirement of one network adapter. You must make a decision regarding the total number of network adapters for the cluster. Provide additional network cards to separate cluster and noncluster-related traffic and to provide required throughput for the host.
11. Identify any special network considerations for the cluster. Special considerations could include a switched network or proxy servers. Select any special considerations that you must give to the cluster configuration because of network configuration.
12. For example, if you use a switched network, you must ensure that you can support multicast protocols, or configure the cluster to use unicast mode.

► To complete Part III Physical Cluster Configuration

1. Different hosts within the cluster can have a different number of network adapters, but all of them must use the same network IP transmission mode, multicast or unicast. The default setting is unicast. All cluster hosts must be either unicast or multicast for the cluster to function properly.
2. Identify the cluster's full Internet name. You use this URL to create a unique signature to allow cluster members to identify heartbeat communications.
3. Select the cluster's virtual IP address.
4. Select the subnet mask for the virtual IP for the cluster.
5. Select the dedicated IP address for each host in the cluster. The dedicated IP address will typically be the host IP address that was used prior to becoming a cluster member.
6. Select the subnet mask for the dedicated IP for each host in the cluster.
7. Select the priority for each host in the cluster. You use the priority to define the failover order for cluster members.

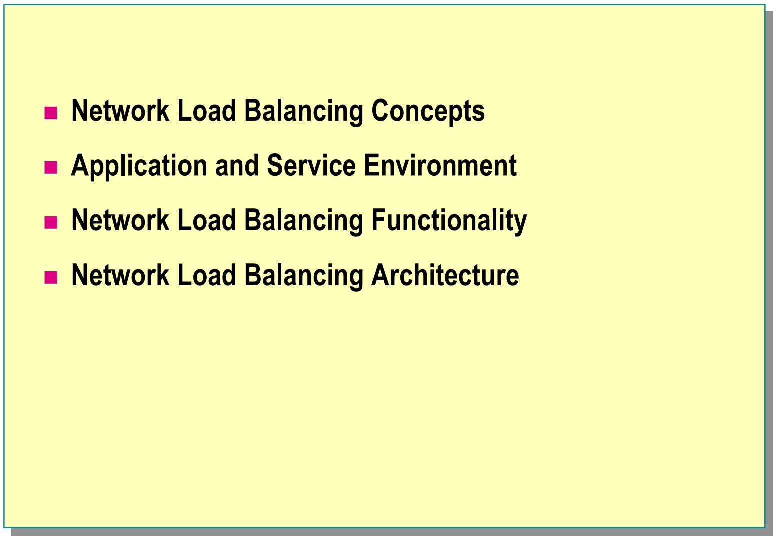
► To complete Part IV Cluster Traffic Handling Configuration

1. Select the required port range (minimum, maximum) for each rule. The number of port rules that are required will depend on the applications and services that are being supported.
2. Select TCP, UDP, or both for the supported protocols for this rule.
3. Select the Filtering mode for inbound traffic, depending on whether you require load balance of failover response.
4. Select client affinity based on the client requirements. If client state is an issue, or proxy servers exist on the network, these issues will influence this decision.
5. Select load weight for this host when Filtering is Multiple (percent). You typically use manual load balancing where the cluster members have differing performance levels.
6. Select handling priority for this rule when Filtering is set to Single (1-32). One is the highest priority.

Review

Topic Objective
To reinforce module objectives by reviewing key points.

Lead-in
The review questions cover some of the key concepts taught in the module.



- Network Load Balancing Concepts
- Application and Service Environment
- Network Load Balancing Functionality
- Network Load Balancing Architecture

****************************ILLEGAL FOR NON-TRAINER USE******************************

1. What two system requirements must you meet before you can use Network Load Balancing to load balance applications or services?

   Applications and services in a Network Load Balancing cluster must use TCP/IP as the network protocol and must be associated with a specific TCP or UDP port.

2. Describe how you can configure the Network Load Balancing driver to manage client connections.

   The Network Load Balancing driver allows you to manage the IP traffic by manually configuring the load, distributing the load evenly across all of the cluster hosts or delegating the load to the highest priority host.

3. What is the convergence process and how does it handle incoming network traffic?

   When interhost communication detects a change in the state of the cluster, convergence is invoked. The hosts then exchange communication that determines a new consistent state of the cluster and elects the cluster host with the highest priority as the default host.

4. Describe the changes made to the Network Load Balancing cluster when you enable unicast or multicast modes.

   When unicast mode is enabled, the cluster's MAC address is assigned to the computer's network adapter and the network adapter's built-in MAC address is not used.

   When you enable multicast, Network Load Balancing adds a mulitcast MAC address to the cluster adapters on all of the cluster hosts.

   When multicast mode is enabled, the cluster's MAC address is assigned to the computer's network adapter, but the network adapter's built-in address is retained so that both addresses are used, the first for client-to-cluster traffic and the second for network traffic that is specific to the computer.

5. There are four configuration modes for the Network Load Balancing cluster; single network adapter in unicast mode, single network adapter in multicast mode, multiple network adapters in unicast mode and multiple network adapters in multicast mode. Describe a suitable scenario in which each of these modes would be implemented.

   The single network adapter in unicast mode is suitable where you do not require interhost communication and there is limited dedicated traffic from outside of the cluster subnet to specific cluster hosts.

   The single network adapter in multicast mode is suitable where interhost communication is necessary and there is limited dedicated traffic from outside the cluster subnet to specific cluster hosts.

   The multiple network adapter in unicast mode is suitable where interhost is necessary and there is relatively heavy dedicated traffic from outside of the cluster subnet to cluster hosts.

   The multiple network adapter in multicast mode is suitable where interhost is necessary and there is heavy dedicated traffic from outside the cluster subnet to cluster hosts.

6. Describe the function of the primary IP and the dedicated IP addresses within the Network Load Balancing cluster.

   The primary IP address is the virtual IP address of the cluster and you must set it identically for all of the hosts in the cluster. You use it to address the cluster and it is associated with the Internet name that you specify for the cluster.

   The dedicated IP address specifies a host's unique IP address and is used for traffic not associated with the cluster. The Network Load Balancing driver does not load balance IP traffic for dedicated IP addresses.