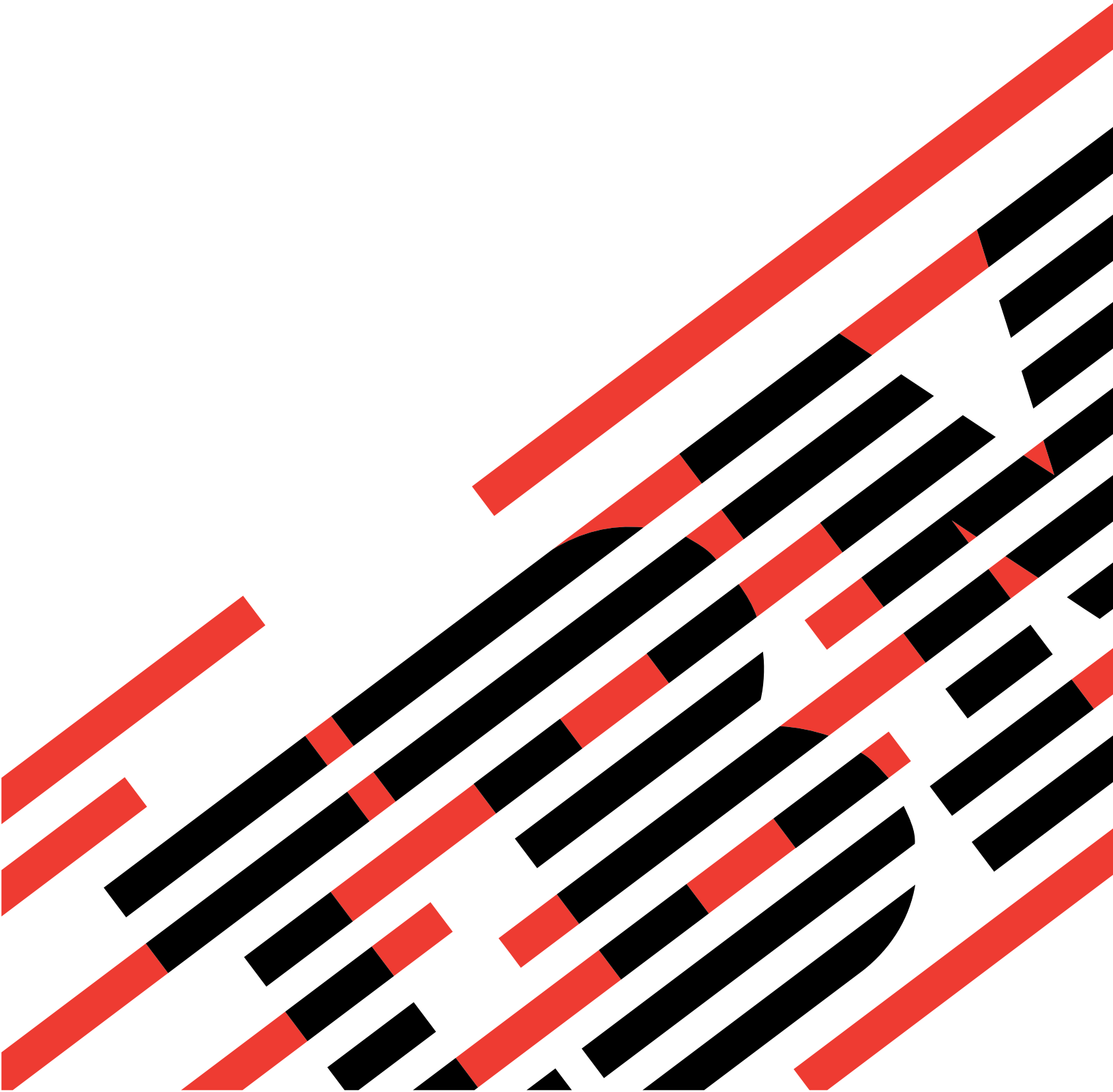
IBM

eserver

iSeries

Quality of service



IBM

@server

iSeries

Quality of service

Contents

Quality of service (QoS)

All traffic in your network receives equal priority. Noncritical browser traffic is considered as important as critical business applications. If your chief executive officer (CEO) is giving a presentation using an audio/video application, IP packet priority becomes a concern. It is critical that, during the presentation, this application receive greater performance than other applications.

QoS allows you to request network priority and bandwidth for TCP/IP applications. Packet priority is important to you if you send applications that need predictable and reliable results, such as multimedia.

It is important to understand QoS before you start planning for policy rules. The following links provide you with the information you need to implement QoS.

What's new for V5R2?
Lists changes to the quality of service networking function and information center topic.

Print this topic
Print this entire topic.

QoS scenarios
View some QoS policy scenarios to see why and how you can use QoS.

QoS concepts
If you are new to quality of service, view some basic QoS concepts and mechanisms. This will give you an overview of how QoS works and how QoS mechanisms work together.

Plan for QoS
Links you to a planning advisor and network information you will need to know in order to use QoS effectively.

Configure QoS
Follow these procedures to create new differentiated service policies and integrated service policies.

Manage QoS
Follow these procedures to edit your existing policies. These articles tell you where to find actual tasks for deleting, tracing, and using other policy management techniques.

Troubleshoot QoS
Use this troubleshooting section to help you debug a QoS problem.

Related information for QoS
Find links to other useful QoS sources. There are many other books, Web sites, request for comments (RFCs) and white papers.

What's new in V5R2?

This article describes new function added for Version 5 Release 2. It also highlights some of the topic's design improvements.

New function

- **Associate policies with local interfaces**
  You can associate policies with a specific local interface or a range of local interfaces on the iSeries(TM). Designating the local interface allows different polices to be based on what interface the client packet arrives on.
- **Associate policies to multiple clients**
  You can now associate a policy to multiple clients. This allows you to create more flexible policy definitions.

- **Inbound admission policies**
  You can now create policies to control external traffic attempting to access your server. There are two new wizards that allow you to control traffic attempting to access a particular IP address or URI value within your network. Use the link above to learn more about the two inbound policies.
- **Monitor information can be stored and printed**
  You can now save and print monitor information. When you save the information, it will be accessible for future reference. If you want to print monitor information, you can now specify, “Export as HTML”.
- **Policies stored on LDAP directory server**
  Policies are now exported to a directory server with the latest LDAP protocol version 3. Using a directory server makes your QoS solution easier to manage. Instead of configuring the same QoS policies on each of your servers, you can configure your servers to use policy data created by a single server. The policies are then saved on the directory server. Use this link to obtain more detail on configuration.
- **Schedule change**
  Schedules are defined by time ranges. In the past a time range had to exist within the same day. Now, a time range can span any twenty-four hour period, even if it overlaps days. Schedules are associated to policies to specify when the policy should be active. This allows you to create more flexible policy definitions.

New design improvements

- **QoS planning advisor**
  The QoS planning advisor has been updated to give you suggestions and prerequisites before configuring the policies. It is used to help plan by bringing the concepts together in an organized location.
- **New inbound scenario**
  A new scenario has been added to show an example of the inbound policy implementation.

How to see what’s new or changed

To help you see where technical changes have been made, this information uses:

- The

  image to mark where new or changed information begins.
- The

  image to mark where new or changed information ends.

To find other information about what’s new or changed this release, see the Memo to Users

.

Print this topic

To view or download the PDF version, select Quality of service (about 378 KB or 53 pages).

To save a PDF on your workstation for viewing or printing:

1. Open the PDF in your browser (click the link above).
2. In the menu of your browser, click **File**.
3. Click **Save As...**
4. Navigate to the directory in which you would like to save the PDF.

5. Click **Save**.

If you need Adobe Acrobat Reader to view or print these PDFs, you can download a copy from the Adobe Web site

.

QoS scenarios

One of the best ways to learn about quality of service is to see how the function works in your overall network picture. The following basic examples show why you would use quality of service policies.

Scenario: Dedicated delivery (IP telephony)
If you need dedicated delivery and want to request a reservation, you use an integrated service policy. There are two types of integrated service policies to create: Guaranteed and controlled load. In this example, we use guaranteed service.

Scenario: Limit browser traffic
You can use QoS to control traffic performance. Use a differentiated service policy to either limit or extend an application's performance within your network.

Scenario: Limit inbound connections
If you need to control the inbound connection requests made to your server, use an inbound admission policy.

Scenario: Predictable B2B traffic
If you need predictable delivery and still need to request a reservation, you also use an integrated service policy. However, this example uses a controlled load service.

Scenario: Secure and predictable results (VPN and QoS)
If you are using a virtual private network (VPN), you can still create quality of service policies. This example shows the two being used together.

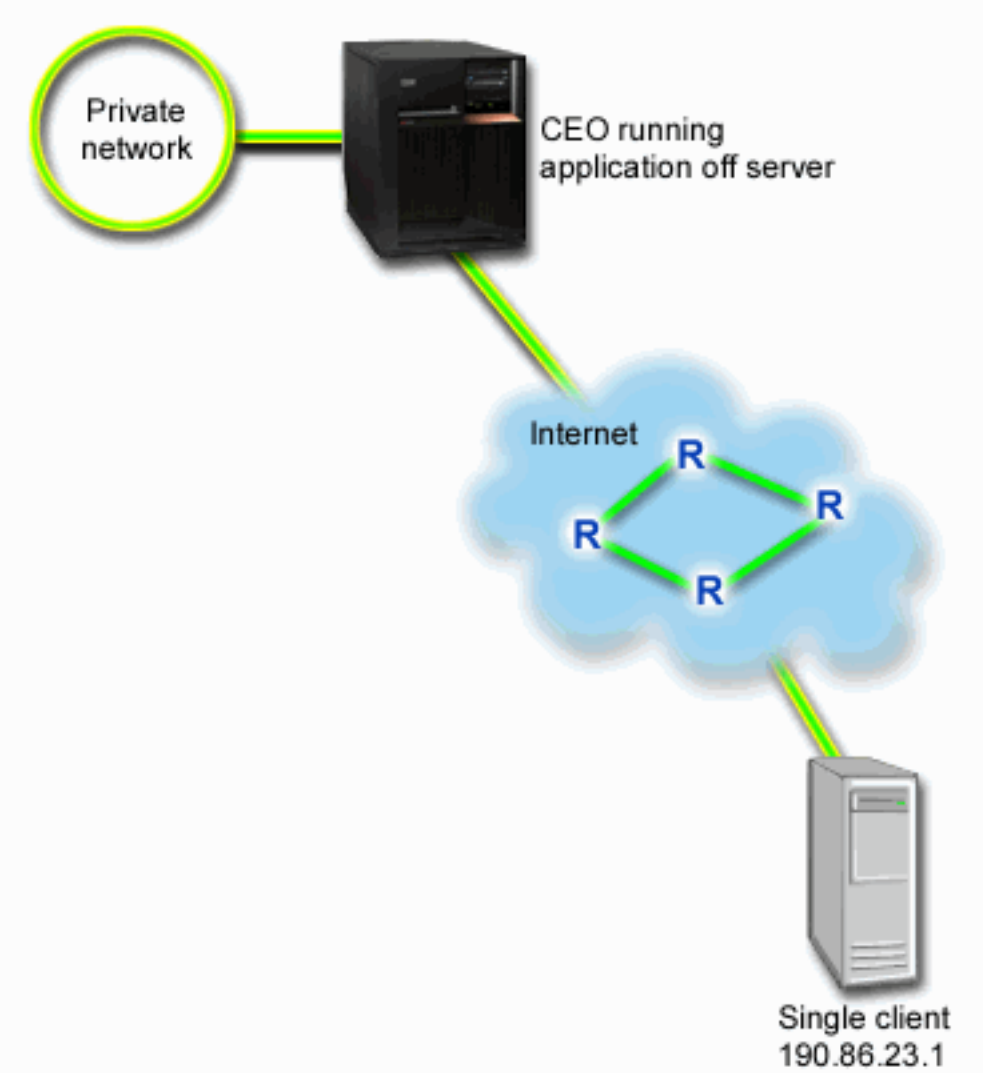
Note: The IP addresses and diagrams are fictitious and only used for example purposes.

QoS scenario: Dedicated delivery (IP telephony)

Problem

The chief executive officer (CEO) of your company is going to give a live broadcast to a client across the country between 1:00PM-2:00PM. You must guarantee that IP telephony will have guaranteed bandwidth so there are no interruptions during the broadcast. In this scenario, the application resides on the server. The following figure illustrates the network setup in this scenario. Your iSeries server is running on OS/400(R) V5R2.

Figure 1. CEO to client presentation guaranteed by an integrated service policy.



Solution

Extremely sensitive applications require a guaranteed connection. Since the application your CEO is using requires a smooth, uninterrupted transfer, you decide to use a guaranteed integrated service policy. Guaranteed service controls the maximum queuing delay, so that packets will not be delayed over a designated amount of time.

Since you want to guarantee this connection, you could use an integrated service policy with guaranteed service. Integrated service policies require RSVP-enabled applications. Since your server does not have any RSVP-enabled applications, you must write your own RSVP-enabled applications. To write your own applications, use the Resource Reservation Setup Protocol (RAPI) API or qtoq QoS socket APIs.

Integrated service policies also require, that along the traffic's path, the routers are RSVP-enabled. See the Integrated service concept section for more information.

Configuration

1. Open QoS within iSeries Navigator.
    1. In iSeries Navigator, expand your server —> **Network—> IP Policies**.

2. Right-click **Quality of Service** and select **Configuration**.
3. Expand **Outbound bandwidth policies**.
4. Right-click **IntServ** and select **New Policy**. The New IntServ policy wizard appears.

2. Create the integrated service policy.

   Your first step is to complete the integrated service policy wizard. Since you want to guarantee traffic from the chief executive officer (CEO), you might call the policy **CEO_guaranteed**. A single client is receiving this presentation at IP address **190.86.23.1**. This is a fictitious number used for example purposes only. You could name this client, **Branch1**. Since this traffic runs on port 2427, you could name the application **port 2427**. You could name the schedule **1:00-2:00**. Use the following values throughout the wizard:

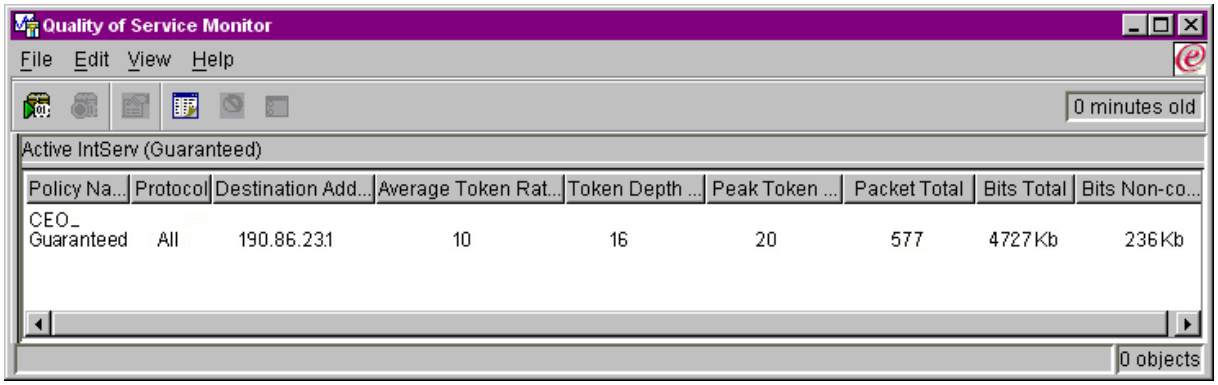
   Name = CEO_guaranteed
   Client = Branch1
   Application = port 2427 (if this is the port that IP telephony is running across)
   Local IP address = 10.5.27.1
   Protocol = TCP
   Schedule = 1:00-2:00
   Token bucket size = 16 Kilobits
   Bandwidth limit (R) = 10 Megabits per second
   Number of flows = 1

   iSeries Navigator lists all the integrated service policies created on your server.

4. Use the monitor to verify your policy is working.
   1. Select the specific Policies folder (DiffServ, IntServ, Server request—>URI, or connection rate).
   2. Right-click the policy that you want to monitor and select **Monitor**.

   Below is a dialog of the monitor output with comments to help explain the results.

   Figure 2. Quality of Service Monitor.



   The most interesting fields are the measured fields that obtain their data from your traffic. These fields include the total bits, bits conformant, and packets conformant. Bits non-conformant would indicate that other traffic is getting delayed or dropped to satisfy this integrated service policy requirements. See the monitor section for a description of all the monitor fields.

3. Modify any values that need adjusting.

   After you view the monitor results for this policy, you can modify the values you previously created with the wizard.

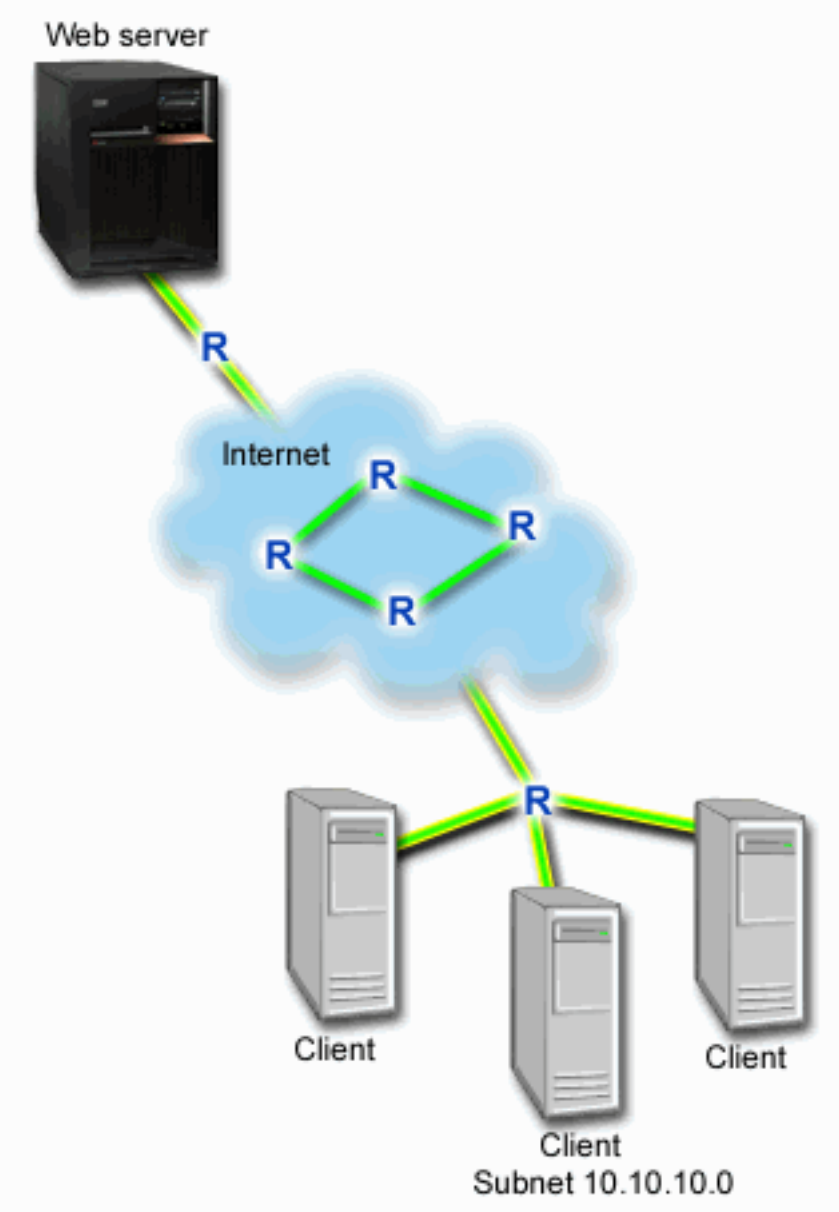
1. Close the monitor.
2. Right-click on the policy name you created above.
3. Select **Properties**, the IntServ_Guaranteed Properties dialog appears.
4. Select the **flow control** tab to change the values that control your traffic's flow. This is also where you edit the schedule, client, applications, and traffic management.

QoS scenario: Limit browser traffic

Problem

Your company has been experiencing high levels of browser traffic from the user-centered design (UCD) group on Fridays. This traffic has been interfering with the accounting department, which also requires good performance from their accounting applications on Fridays. You decide to limit browser traffic from the UCD group. The following figure illustrates the network setup in this scenario. Your iSeries server is running on OS/400(R) V5R2.

Figure 3. Web server limiting browser traffic to a client.



Solution

To limit browser traffic out of your network, you could create a differentiated service policy. A differentiated service policy divides your traffic into classes. All traffic within this policy is assigned a codepoint. This codepoint tells routers how to treat the traffic. In this scenario, the policy would be assigned a low codepoint value to affect how the network prioritizes browser traffic.

Configuration

1. Open QoS within iSeries Navigator.
    1. In iSeries Navigator, expand your server —> **Network—> IP Policies**.
    2. Right-click **Quality of Service** and select **Configuration**.
    3. Expand **Outbound bandwidth policies**.
    4. Right-click **DiffServ** and select **New Policy**. The new DiffServ policy wizard appears.

2. Create the differentiated service policy.

   Since you want to limit browser traffic to the user-centered design (UCD) group, you might call the policy **UCD**. The clients use a subnet address of **10.10.10.0**. This is a fictitious number used for example purposes only. Web traffic generally runs on port 80, so you could name the application **port 80**. Since the congestion only occurs on Fridays, you could apply a 9:00 AM-5:00 PM schedule to the policy. You could name this **Friday9-5**. Use the following settings throughout the wizard:

   Name = UCD (can be any name you assign)
   Client = Subnet 10.10.10.0
   Application = port 80 (well-known port for HTTP traffic)
   Protocol = TCP
   Schedule = Fridays9-5

   Enter the rest of the policy information from the Class of service wizard that, as you continue, will automatically appear.

   Token bucket size = 8 Kilobits
   Average rate limit = 10 Megabits per second
   Peak rate limit = 20 Megabits per second
   Out-of-profile traffic overflow handling = Drop packets (retransmitted)

   iSeries Navigator lists all the differentiated service policies created on your server. After you complete the wizard, your policy is listed in the right-hand pane.

3. Complete a new class of service.

   While completing the wizard, you are asked to assign a per-hop behavior, performance limits, and out-of-profiling traffic handling. This is defined in a class of service.

   Classes of service actually determine the performance levels that this traffic receives from a router. You could name your class of service **Bronze**, to show that this traffic receives a lower service. iSeries Navigator lists all the classes of service defined on your server.

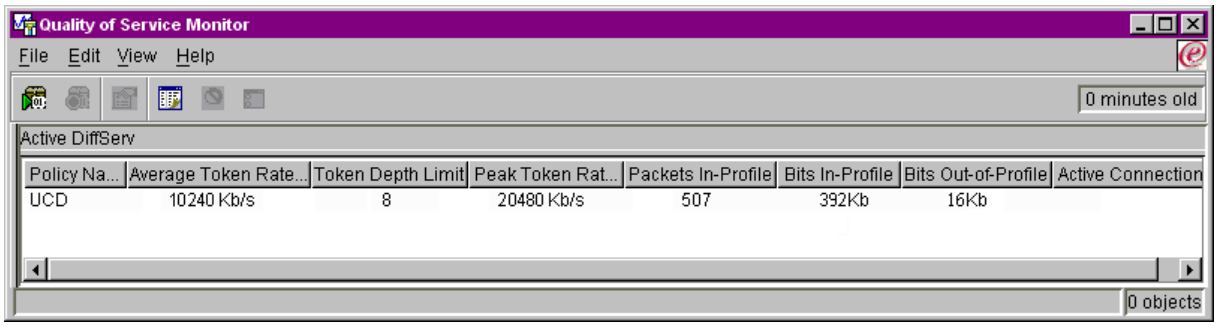
   Class of service name = Bronze

4. Use the monitor to verify your policies are working.

   To verify that the policy is behaving as you configured in the policy, use the monitor.
   1. Select the specific Policies folder (DiffServ, IntServ, Server request—>URI, or connection rate).
   2. Right-click the policy that you want to monitor and select **Monitor**.

   Below is a dialog of the monitor output with comments to help explain the results.

   Figure 4. Quality of Service Monitor.



The most interesting fields are the fields that obtain their data from your traffic. Make sure to check the total bits, bits in-profile, and packets in-profile fields. Bits out-of-profile would indicate when traffic exceeds the configured policy values. In a differentiated service policy, the out-of-profile number indicates the number of bits being dropped. The in-profile packets indicate the number of bits controlled by this policy (from the time the packet was started to the present monitor output).

The value you assign the average rate limit field is also important. When packets exceed this limit, the server will begin to drop them. As a result, the bits out-of-profile will increase. This shows you that the policy is behaving as you configured it to behave. See the monitor section for a description of all the monitor fields.

5. Change any values that do not apply to this policy.

   You can modify any of the values you created in the policy.

   1. Close the monitor.
   2. Select Classes of Service in the left pane.
   3. In the right pane, right-click on the class of service name you created above.
   4. Select **Properties**. A CoS Properties dialog appears with the values that control your traffic. Modify the appropriate values.

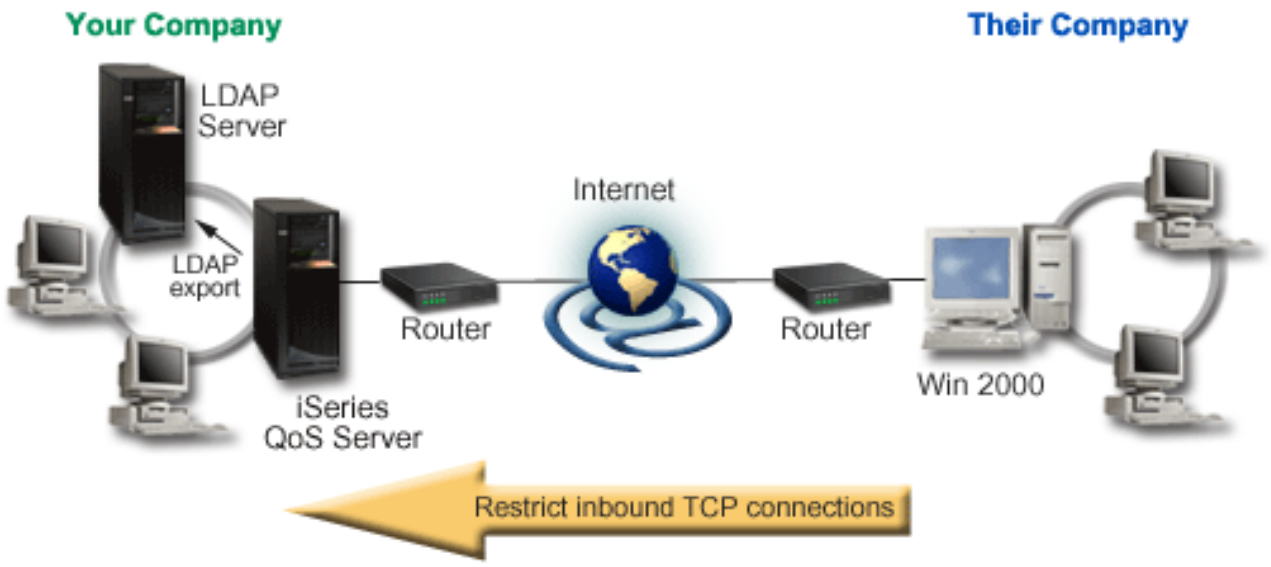
QoS scenario: Limit inbound connections

Problem

Your web server's resources are being overloaded by client requests entering your network. You are asked to slow incoming HTTP traffic to your web server (10.1.1.4) on the local interface 10.1.1.1. QoS can help you restrict the accepted inbound connection attempts, based on connection attributes (For example, IP address) to your server. To achieve this, you decide to implement an inbound admission policy, which will restrict the number of accepted inbound connections.

The illustration shows your company and a client company. This QoS policy can only control traffic flow in one direction.

Figure 5. Restricting inbound TCP connections.



Prerequisites:

- Running iSeries V5R2
- Configured and running LDAP server

Solution

To configure an inbound policy, you must decide whether you are restricting traffic to a local interface or a specific application and whether or not you are restricting it from a particular client. In this case, you want to create a policy that restricts connection attempts from Their_Company going to port 80 (HTTP protocol) on your local interface 10.1.1.1. Since you are defining this restriction by IP address, you should create a Connection rate policy. There are two types of inbound admission policies: Connection rate and Server request (URI). URI policies restrict connections attempting to access either a particular relative URI name (similar to the relative URL) or all URLs on your system. If you want more information on the URI policy, see Inbound admission policies.

To create this connection rate policy and complete the scenario above, open iSeries Navigator and go to the QoS function.

Configuration

1. Open QoS within iSeries Navigator.
    1. In iSeries Navigator, expand your server —> **Network—> IP Policies**.
    2. Right-click **Quality of Service** and select **Configuration**.
    3. Expand **Inbound admission policies**.
    4. Right-click **Connection rate** and select **New Policy**.

2. Complete the Connection rate policy wizard.

    Your second step is to complete the new connection rate policy wizard. Since you want to restrict traffic from Their_Company, you might call the policy **Restrict_TheirCompany**. You want to restrict requests made to your local IP address of 10.1.1.1 from the client, Their_Company. This is a fictitious

number used for example purposes only. Since this traffic runs on port 80, you could name the application **port 80**. You could name the schedule **Weekdays(9-5)**. Use the following values throughout the wizard:

Name = Restrict_TheirCompany
Client = Their_Company
Application = port 80
Local IP address = 10.1.1.1
Schedule = Weekdays (9-5)
Average connection rate = 100 per second
Connection burst rate = 5 connections
Priority = Medium

iSeries Navigator lists all the connection rate policies created on your server.

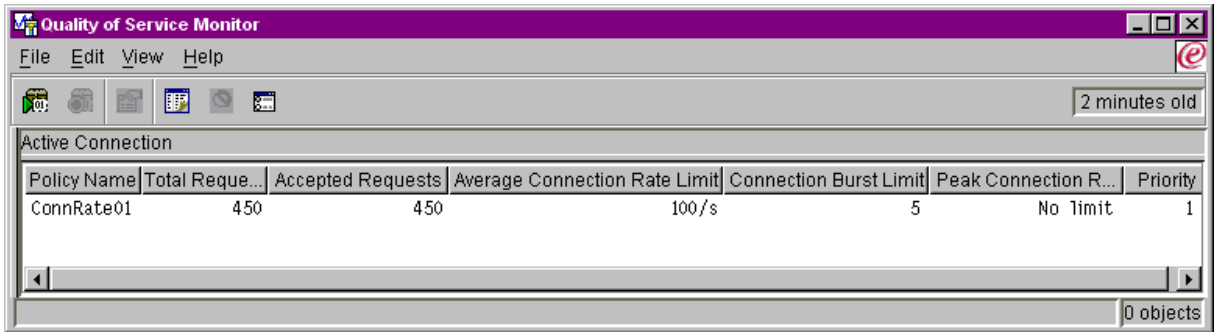
3. Monitor the traffic contained in this policy, to make sure you see the results you expect.

   To verify that the policy is behaving as you configured it to behave, use the monitor.

   1. Select the specific Policies folder (DiffServ, IntServ, Server request—>URI, or connection rate).
   2. Right-click the policy that you want to monitor and select **Monitor**.

   Below is a figure of the monitor output with comments to help explain the results.

   Figure 6. Quality of Service Monitor.



   Make sure to check any measured fields, such as accepted requests, dropped requests, total requests, and connection rate. Dropped requests would indicate when traffic exceeds the configured policy values. The accepted requests indicate the number of bits controlled by this policy (from the time the packet was started to the present monitor output).

   The value you assign the average connection request rate field is also important. When packets exceed this limit, the server will begin to drop them. As a result, the dropped requests will increase. This shows you that the policy is behaving as you configured it to behave. See the monitor section for a description of all the monitor fields.

4. If you need to modify any values, change them in the properties panels.

   Close the monitor. **Right-click** the Restrict_TheirCompany policy and select **Properties**. These panels allow you to edit the policy's properties. This is also where you edit the schedule, client, applications, and traffic management.

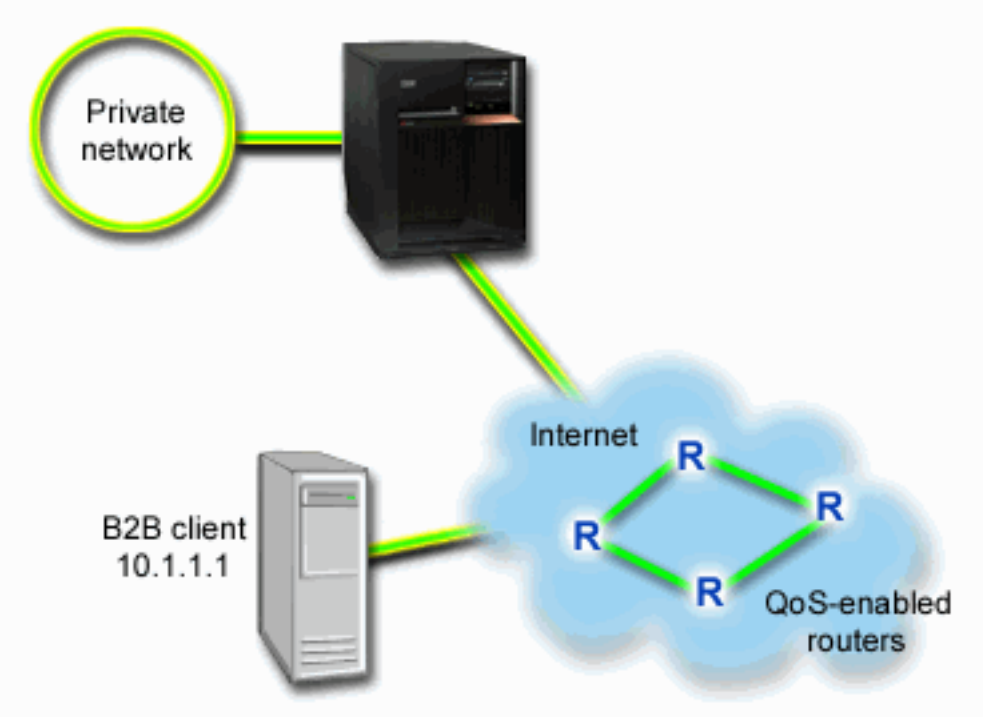
QoS scenario: Predictable B2B traffic

Problem

The Sales department reports problems that network traffic is not performing as they expected. Your company's iSeries server resides in a business-to-business (B2B) environment that requires predictable e-business service. You need to provide predictable transactions to your customers. You want to give the sales unit a higher quality of service for their ordering application during the busiest time of the day (between 10 AM and 4 PM).

In the illustration below, the sales team is within your private network. There are RSVP-enabled routers along the traffic's path to the B2B client. Each `R` represents a router along the traffic's path.

Figure 7. Integrated services policy to a B2B client using RSVP-enabled routers.



Solution

Controlled load service supports applications that are highly sensitive to congested networks, but are still tolerant to small amounts of loss and delay. If an application uses the controlled load service, its performance will not suffer as network load increases. Traffic will be provided with service resembling normal traffic in a network under light conditions. Since this particular application is tolerant to some delay, you decide to use an integrated services policy using a controlled load service.

Integrated service policies require RSVP-enabled applications. Since your server does not have any RSVP-enabled applications, you must write your own RSVP-enabled applications. To write your own applications, use the Resource Reservation Setup Protocol (RAPI) API or qtoq QoS socket APIs.

Integrated service policies also require that, along the traffic's path, the routers are RSVP-enabled. See the Integrated services concept section for more information.

Configuration

1. Open QoS within iSeries Navigator.
    1. In iSeries Navigator, expand your server —> **Network—> IP Policies**.
    2. Right-click **Quality of Service** and select **Configuration**.
    3. Expand **Outbound bandwidth policies**.
    4. Right-click **IntServ** and select **New Policy**. The new IntServ policy wizard appears.

2. Create a new integrated service policy.

    Since you want to give predictable traffic to your customers, you might call the policy **B2B_CL**. A single client is receiving this transaction at IP address **10.1.1.1**. This is a fictitious number used for example purposes only. Since this traffic runs on various ports between 7000 and 8000, you could name the application **port 7000-8000**. Since this transaction is occurring from 10:00-4:00, you could name the schedule **Primetime**. Use the following settings throughout the wizard:

    Name = B2B_CL
    Client =10.1.1.1
    Application = port 7000-8000
    Protocol = TCP
    Schedule = Primetime
    Token bucket size (b) = 8 Kilobits
    Token rate limit = 25 Megabits per second
    Token bucket size (r) = 75 Kilobits
    Number of flows = 5

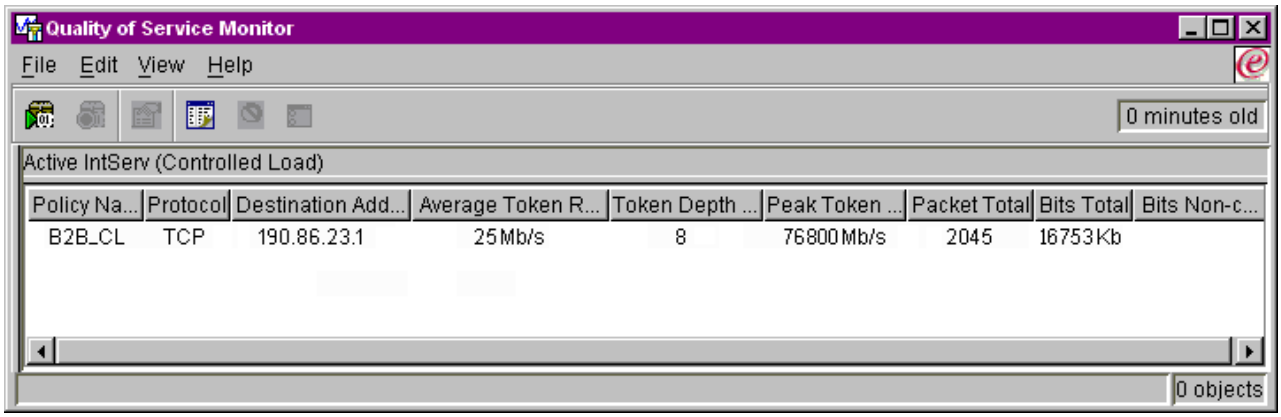
    iSeries Navigator lists all the integrated services policies created on your server.

3. Use the monitor to verify your policies are working.

    To verify that the policy is operating correctly, use the monitor.
    1. Select the specific Policies folder (DiffServ, IntServ, Server request—>URI, or connection rate).
    2. Right-click the policy that you want to monitor and select **Monitor**.

    Below is a dialog of the monitor output with comments to help explain the results.

Figure 8. Quality of Service Monitor.



The most interesting fields are the fields that obtain their data from your traffic. Make sure to check the total bits, bits conformant, and packets conformant fields. Bits non-conformant would indicate that other traffic is getting delayed or dropped to satisfy this integrated services policy requirements. For a full description of the monitor fields, see the monitor section.

4. Modify any values that need adjustment within this policy.

After you create this policy, you can modify the values you previously created with the wizard.

1. Close the monitor.
2. Right-click on the policy name you created above.
3. Select **Properties**, the B2B_CL Properties dialog appears.
4. Select the **flow control** tab to change the values that control your traffic's flow.

This is also where you edit the schedule, client, applications, and traffic management.

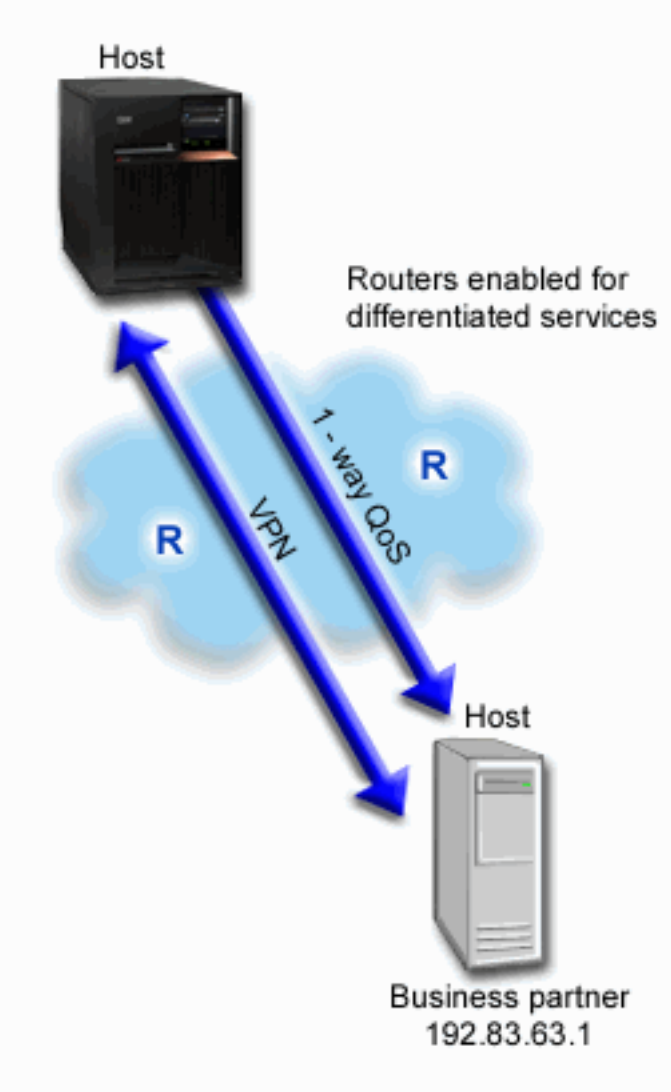
QoS scenario: Secure and predictable results (VPN and QoS)

Problem

You have a business partner connected through a VPN and you want to combine VPN and QoS to provide security and predictable e-business flow for mission-critical data. The QoS configuration only travels in one direction. Therefore, if you have an audio/video application, you need to establish QoS for the application on both sides of the connection.

The illustration shows your server and your client in a host-to-host VPN connection. Each R represents differentiated service-enabled routers along the traffic's pathway. As you can see, QoS policies only flow in one direction.

Figure 9. Host-to-host VPN connection using a QoS differentiated service policy.



Solution

You would use VPN and QoS to establish not only protection, but priority for this connection. First, you would need to set up a host-to-host VPN connection. See the Host-to-Host VPN connection example, to assist you with the VPN configuration. Once you have the protection of your VPN connection, you can set up your QoS policy. You could create a differentiated service policy. This policy would be assigned a high expedited forwarding codepoint value to affect how the network prioritizes mission-critical traffic.

Configuration

1. Set up a host-to-host VPN connection. See the Host-to-Host VPN connection example, to assist you with the VPN configuration.

2. Open QoS within iSeries Navigator.
    1. In iSeries Navigator, expand your server —> **Network—> IP Policies**.
    2. Right-click **Quality of Service** and select **Configuration**.
    3. Expand **Outbound bandwidth policies**.

4. Right-click **DiffServ** and select **New Policy**. The new DiffServ policy wizard appears.

3. Create the differentiated service policy.

   Since you want to increase performance for B2B applications, you might call the policy **B2B**. The clients have a single address of **192.83.63.1**. This is a fictitious number used for example purposes only. Since B2B traffic can use any port, you could name the application **All ports**. Since the congestion only occurs between 9:00 AM and 5:00 PM, we could apply a 9-5 schedule to the policy. You could name this **Firstshift**. Use the following settings throughout the wizard:

   Name = B2B
   Client = VPNClient
   Application = All port
   Protocol = All
   Schedule = Firstshift

   Enter the rest of the policy information from the Class of service wizard that, as you continue, will automatically appear.

   Token bucket size = 8 Kilobits
   Average rate limit = 90 Megabits per second
   Peak rate limit = Do not limit
   Out-of-profile traffic overflow handling = Drop packets (retransmitted)

   iSeries Navigator lists all the differentiated service policies created on your server.

4. Complete a new class of service.

   While completing the wizard, you will be asked to assign a class of service. The class of service assigns performance limits, codepoints, and out-of-profile handling characteristics. In this policy, you will want to assign a high priority, expedited forwarding codepoint. Since you want to apply an expedited forwarding codepoint, you could name the class of service **EF_VPN** to remind yourself why you selected this value.

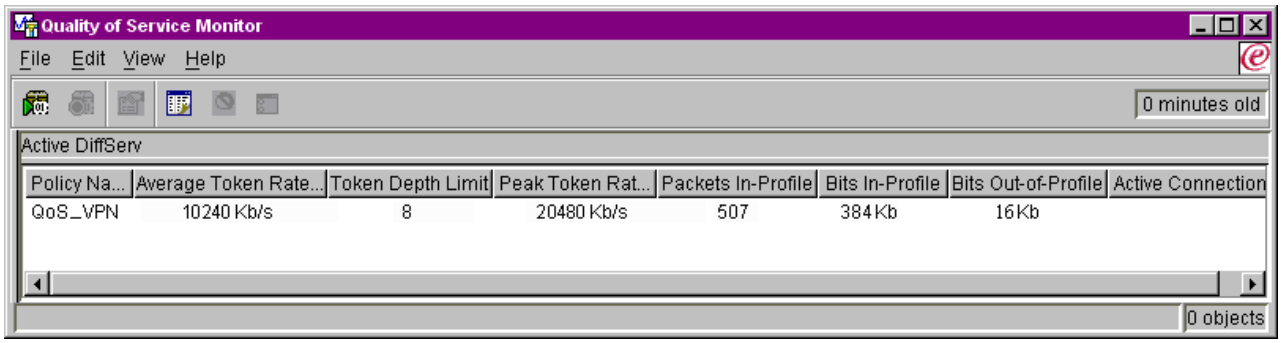
   Class of service = EF_VPN

6. Use the monitor to verify your policies are working.

   To verify that the policy is behaving as you configured it to behave, use the monitor.
   1. Select the specific Policies folder (DiffServ, IntServ, Server request—>URI, or connection rate).
   2. Right-click the policy that you want to monitor and select **Monitor**.

   Below is a figure of the monitor output with comments to help explain the results.

Figure 10. Quality of Service Monitor.



Similar to example 1, the most interesting fields are the fields that obtain their data from your traffic. These fields include the total bits, bits conformant, and packets conformant fields. Bits non-conformant would indicate when traffic exceeds the configured policy values. The conformant packets indicate the number of packets controlled by this policy. What values you assign the average rate limit field is very important. When packets exceed this limit, the server will begin to drop them. As a result, the bits non-conformant will increase. The difference between this policy and the example 1 is that the packets here are protected using the VPN protocols. As you can see, QoS does work with a VPN connection. See the monitor section for a description of all the monitor fields.

5. Modify any values that need adjustment within this policy.

   You can also edit the class of service after you create it.

   1. Close the monitor.
   2. Select **Classes of Service** in the left pane.
   3. In the right pane, right-click on the class of service name you created above.
   4. Select **Properties**. A CoS Properties dialog appears with the values that control your traffic. Modify the appropriate values.

QoS concepts

Quality of service (QoS) terms can be found in multiple sources, so this topic will only cover the basics as they specifically apply to your iSeries server.

One of the most important parts of implementing quality of service is your server itself. Not only do you need to understand the concepts below, but you also need to be aware of the role your server plays in implementing these concepts. The iSeries server can only act as a client or a server, not a router. You need to take this into consideration as you learn more about the concepts below and begin planning for quality of service.

To implement QoS, you create policies for your traffic. A policy is a set of rules that designate an action. The policy basically states what client, application, and schedule (which you designate) should receive a particular service. You can ultimately implement four policy types. The policies are first broken into two categories: outbound bandwidth and inbound admission. Within outbound bandwidth policies you can create two service types: integrated service policy or differentiated service policy. Within the Inbound admission policies you can also create two service types: new connection request rate policies and new URI request rate policies.

Inbound refers to policies that control connection requests coming into your network from some outside source. Outbound refers to policies that place limits or help benefit traffic trying to leave your network. To decide which policy you need to use, evaluate the reason why you want to use QoS. Research the concepts below to find out what situations would be suited for each policy type.

Use the following links for more information:

Differentiated services
This is the first type of outbound bandwidth policy you can create on your server. Differentiated services is the portion of QoS that divides your traffic into classes. To implement quality of service in your network, you need to determine how you want to classify your network traffic and how to handle the different classes. Then you can create the classes of service to use with your differentiated services policy.

Differentiated classes of service
This subtopic explains the parts that make up a class of service. When you create a differentiated services policy, you also have to create a class of service.

Integrated services
The second type of outbound bandwidth policy you can create is an integrated service policy. Integrated services provides the capability for IP applications to request and reserve bandwidth using the RSVP protocol. Integrated service policies use the RSVP protocol to guarantee an end-to-end connection. This is the highest level of service you can designate; however, it is also the most complex. When you create an integrated service policy, you will designate one of two service classes: guaranteed service or controlled load service.

Integrated services using differentiated services markings
This type of policy is generally used when an integrated services policy may cross a mixed network environment. A mixed network environment contains some network nodes that are RSVP-enabled and some that are not RSVP-enabled.

RSVP and the QoS APIs
This subtopic describes the protocol and APIs used to make an integrated services reservation. It also discusses what makes a router RSVP-enabled.

Connection rate
This type of inbound policy is used to control traffic requesting admission (by IP address) into your network. There are two types of inbound admission policies: connection rate and URI. This topic describes both types of inbound policies.

URI
This type of inbound policy is used to control traffic requesting admission (by URI) into your network. There are two types of inbound admission policies: connection rate and URI. This topic describes both types of inbound policies.

Directory server
QoS policies are now exported to a directory server. View this topic to see the benefits of using a directory server, LDAP concepts and configuration, as well as the QoS schema.

Before attempting to implement QoS, you should research quality of service in depth and make sure this service will meet your needs. See the related information for QoS page for help on finding additional resources.

Connection request rate and URI request rate

Inbound policies are used to control traffic attempting to connect to your server. There are two types of policies that allow you to define and configure inbound controls: URI policies and connection rate policies. The two policy types are described below.

URI request rate policies
URI request rate policies are part of a solution to help protect servers against overload. This type of policy applies admission controls, based on application level information, to limit the URI requests accepted by the server. In industry this is also referred to as *header-based connection request control*, which uses URIs to set priorities.

Unlike connection rate policies, URI policies have more control since they examine content, not just packet headers. The content they examine could include URI name or other application specific information. For iSeries, the relative URI name is used to define the policy. For example, **`/products/clothing`**. The examples below describe the relative URI.

Relative URI
The relative URI is actually a subset of an absolute URI (similar to the old absolute URL). Consider this example: http://www.ibm.com/software. The **http://www.ibm.com/software** segment is considered the absolute URI. The **/software** segment is the relative URI. All relative URI values must begin with one forward slash (`/`). The following are valid relative URI examples:

- /market/grocery#D5
- /software
- /market/grocery?q=green

Note: The default protocol, hostname, and port are all inherited from the HTTP server. Also, there is an implicit wildcard when you specify a URI. For example, **/software** will include anything within the software directory.

URI policies are considered inbound policies because they control the traffic requests entering your network. As part of this inbound control, you can specify the priority in which URI requests are handled after they have been accepted by the policies. By prioritizing policies, you actually prioritize the connection requests in the queue based on the configured priority for each connection.

Connection Rate policies
Connection rate policies are also part of a solution to help protect servers against overload. This type of policy applies admission controls, based on connection level information, to limit the connections accepted by the server. In industry this is also referred to as *TCP SYN policing*.

Connection rate policing accepts or denies new incoming connections based on the average number of connections established per second and the maximum number of established connections (in any given instant) defined in the policy you create. These connection limits consist of average rate and burst limits, which the wizards in iSeries navigator will prompt you to enter. When incoming connection requests reach the server, the server analyzes the packet header information to determine if this traffic is defined in a policy. The system verifies this information against the connection limits profile. If the policy is within the policy limits, it is placed into the queue. Packets that do not comply with a policy are discarded.

Similar to URI policies, connection rate policies are considered inbound policies because they control the connection rate of traffic entering your network. As part of this inbound control, you can specify the priority in which connections are handled after they have been accepted by the policies. By prioritizing policies, you actually prioritize the connection requests in the queue based on the configured priority for each connection.

Both the URI policy and connection rate policy require you to set connection rates and burst limits for the traffic defined in each policy. These rate limits help restrict inbound connections trying to enter your server. The average connection rate specifies the limit of new, established connections or rate of accepted URI requests allowed into a server.

Average connection rate and burst limits

Connection rates and burst limits are together known as rate limits. These rate limits help restrict inbound connections trying to enter your server. Rate limits are set within inbound admission policies, both URI and Connection rate.

Connection burst limit
The burst limit size determines the buffer capacity, which holds bursts of connections. Connection bursts may enter the server at a faster rate than it can handle or that you may want to allow. If the number of connections in a burst exceeds the connection burst rate you set, then the additional connections are discarded.

Average connection rate
The average connection rate specifies the limit of new, established connections or rate of accepted URI requests allowed into a server. If a request would cause the server to exceed the limits you set, the server denies the request. The average connection request limit is measured in connections per second.

Hint: To determine what limits to set, you may want to run the monitor. For your reference, see Monitor current network statistics for a sample policy that will help you collect most data travelling on your server. Using these results, you can adjust the limits appropriately.

Differentiated service

Differentiated services divides your traffic into classes. To implement quality of service in your network, you need to determine how you want to classify your network traffic and how to handle the different classes.

The server uses bits in the IP header to identify an IP packet's level of service. Routers and switches allocate their resources based on the per-hop behavior (PHB) information in the IP header's TOS field. The TOS field was redefined in request for comment (RFC) 1349 and OS/400(R) V5R1. A PHB is the forwarding behavior a packet receives at a network node. It is represented by a hexadecimal value known as a codepoint. Packets can be marked at either the server or other parts of the network, such as a router. For a packet to retain the service requested, every network node must be differentiated services-enabled. That is, the equipment must be able to enforce per-hop behaviors. To enforce PHB treatment, the network node must be able to use queue scheduling and outbound priority management. See the Traffic conditioners page for more information on what it means to be differentiated services-enabled.

If your packet passes through a router or switch that is not differentiated services-enabled, it will lose its level of service. Note that the packet is still handled, but it may experience unexpected delivery. On your iSeries server you can use the standard PHB codepoints or you may define your own class. It is not recommended that you create your own codepoints for use outside your private network.

Unlike integrated services, differentiated services traffic does not require a reservation or per-flow treatment. All traffic placed in the same class is treated equally.

Differentiated services is also used for traffic control into or out of a server. This means that your iSeries server really uses differentiated services to limit performance. Limiting a less-critical application allows a mission-critical application to exit your private network first. When you create a policy, you are asked to set various limits on your server. The performance limits include token bucket size, peak rate limit, and average rate limit. The help topics within the QoS function of iSeries Navigator gives you more specific information on these limits.

Now, you know a little more about using differentiated services to group your traffic. If you do not know which codepoints to assign, review codepoints and per-hop behaviors. If you still do not know which codepoints to use, use trial and error. Create test policies, monitor these policies, and make adjustments accordingly.

Differentiated classes of service

The differentiated services section discusses how the differentiated services function groups your traffic into classes. Even though most of this happens through equipment, you control how you group traffic and what priority the traffic should receive.

As you implement QoS, you will first define policies. The policies determine the who, what, where, and when. Then you must assign a class of service to your policy. Classes of service are defined separately and may be reused by policies. A class of service is comprised of a per-hop behavior, traffic limits, and out-of-profile handling in the class of service.

Per-hop behaviors
Quality of service uses the recommended codepoints to assign per-hop behaviors to traffic. Routers and switches use these codepoints to give traffic priority levels. Your server can not use these codepoints, since it does not act as a router. You should determine which codepoints to use based on your individual network needs. Consider what applications are most important to you and what policies should be assigned higher priority. The most important thing is to be consistent with your markings, so that you get the results you expect. These codepoints will be a key part of differentiating different classes of traffic.

Performance limits
Quality of service uses performance limits to restrict traffic through your network. These limits are placed by setting the token bucket size, peak rate limit, and average rate limit. See Token Bucket and Bandwidth limit for more information about these specific values.

Out-of-profile handling
The final portion of a class of service is out-of-profile handling. When you assign the performance limits above, you set values to restrict traffic. When traffic exceeds these restrictions, the packets are considered out-of-profile. This information in a class of service tells the server whether to drop, shape, or retransmit these out-of-profile packets. If you decide to drop out-of-profile packets, they are retransmitted after a specified amount of time. If you delay the out-of-profile packets, they are shaped to conform to your defined handling characteristics. If you remark out-of-profile packets with a Differentiated Service CodePoint (DSCP), they are reassigned a new codepoint. When you assign these handling instructions in the wizard, click Help for more specific information.

Codepoints and per-hop behaviors

Quality of service uses the following recommended codepoints to assign per-hop behaviors to traffic. You should determine which codepoints to use based on your individual network needs. Only you can decide what codepoint schemes make sense for your environment. You need to consider what applications are most important to you and what policies should be assigned higher priority. The most important thing is to be consistent with your markings, so that you get the results you expect.

This table displays the recommended codepoints. You may also create your own per-hop behaviors.

Expedited forwarding (See 22)	Class selector (See 22)	Assured forwarding (See 22)
101110	Class 0 - 000000	Assured forwarding, Class 1, Low - 001010
	Class 1 - 001000	Assured forwarding, Class 1, Medium - 001100
	Class 2 - 010000	Assured forwarding, Class 1, High- 001110
	Class 3 - 011000	Assured forwarding, Class 2, Low - 010010

	Class 4 - 100000	Assured forwarding, Class 2, Medium - 010100
	Class 5 - 101000	Assured forwarding, Class 2, High - 010110
	Class 6 - 110000	Assured forwarding, Class 3, Low - 011010
	Class 7 - 111000	Assured forwarding, Class 3, Medium - 011100
		Assured forwarding, Class 3, High - 011110
		Assured forwarding, Class 4, Low - 100010
		Assured forwarding, Class 4, Medium - 100100
		Assured forwarding, Class 4, High - 100110

Expedited forwarding
Expedited forwarding is one type of differentiated services per-hop behavior. It is mainly used to provide guaranteed service across a network. Expedited forwarding gives traffic a low-loss, low-jitter, end-to-end service by guaranteeing bandwidth across networks. The reservation is made before the packet is sent. The main goal is to avoid delay and deliver the packet on a timely basis.

Note: There is usually a high cost to receive expedited forwarding treatment, so it is not recommended to use this per-hop behavior on a regular basis.

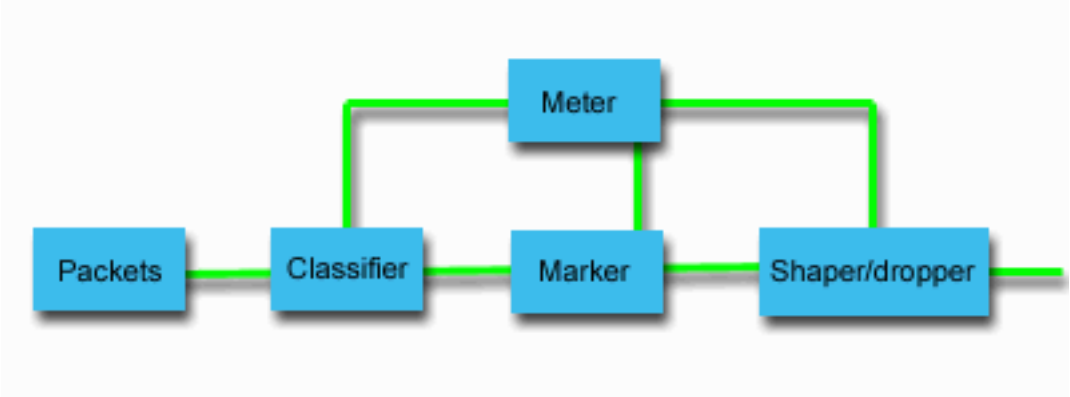
Class selector
Class selector codepoints are another type of differentiated services behavior. There are seven classes. Class 0 gives packets the lowest priority and Class 7 gives packets the highest priority within the class selector codepoint values. This is the most common group of per-hop behaviors, because most routers already use similar codepoints.

Assured forwarding
Assured forwarding is divided into four per-hop behavior classes, which each have drop precedence levels of low, medium, or high. A drop precedence level determines how likely it is for the packets to be dropped. The classes each have their own bandwidth specifications. Class 1, High gives the policy the lowest priority and Class 4, Low gives policies the highest priority. A low drop level means the packets in this policy have the lowest chance of being dropped in this particular class level.

Traffic conditioners
Network equipment using quality of service policies, needs to be QoS-aware. This means that network equipment, such as routers and switches must have the following capabilities: classifiers, meters, markers, shapers, and droppers. The collection of these terms is referred to as *traffic conditioners*. If the network equipment has all the traffic conditioners, then it is considered QoS-aware.

The following figure shows a logical representation of how traffic conditioners work.

Figure 11. Traffic conditioners



The following information describes each of the traffic conditioners in more detail.

Classifiers
Packet classifiers select packets in a traffic stream based on the content in its IP header. The iSeries server defines two types of classifiers. The BA (Behavior aggregate) classifies packets based exclusively on the differentiated services codepoint. The MF (Multi-field) classifier selects packets based on the value of a combination of one or more header fields, such as source address, destination address, differentiated services field, protocol ID, source port, and destination port numbers.

Meters
Traffic meters measure whether or not the IP packets, being forwarded by the classifier, are corresponding to the traffic's IP header profile. The information in the IP header is determined by the values you set in the QoS policy for this traffic. A meter passes information to other conditioning functions to trigger an action. The action is triggered for each packet whether it is in-profile or out-of-profile.

Markers
Packet markers set the differentiated services (DS) field. They take the differentiated services codepoint setting and transfer it into bytes. The marker can be configured to mark all packets to a single codepoint or to a set of codepoints used to select a per-hop behavior.

Shapers
Shapers delay some or all of the packets in a traffic stream to bring the stream into compliance with the traffic profile. A shaper has a finite buffer size, and packets may be discarded if there is not enough space to hold the delayed packets.

Droppers
Droppers discard some or all of the packets in a traffic stream. This occurs to bring the stream into compliance with the traffic profile.

Directory server concepts

QoS policy configurations are stored on an LDAP directory server. You must use an LDAP server with the latest LDAP protocol version 3.

Benefits of using a directory server
Using a directory server makes your QoS solution easier to manage. Instead of configuring QoS policies on all of your servers, you can store the configuration data on one local directory server for many systems to share. However, sharing data is not necessary. There are two other ways to use the directory server with QoS.

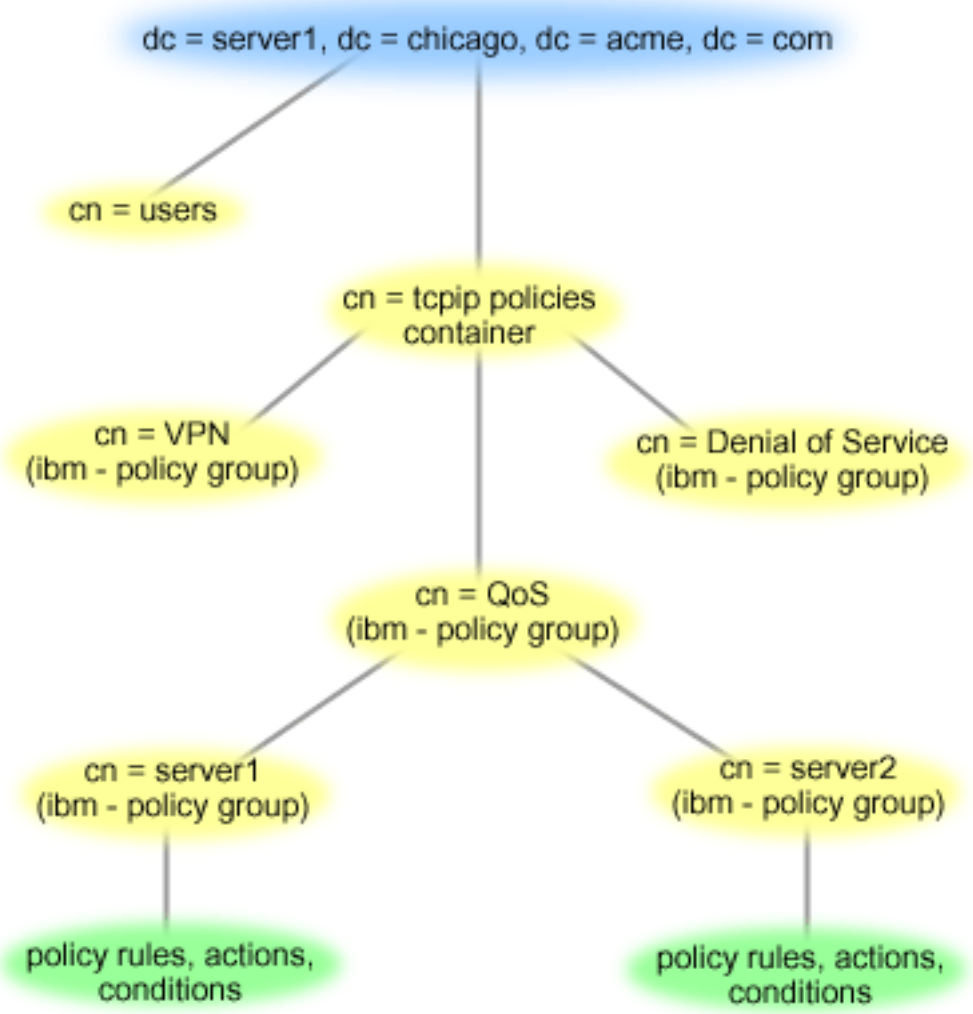
1. Data can still be configured, stored, and only used by one system.

2. The configuration data can also reside on a directory server that holds data for other systems, but is not necessarily shared between those other systems. This allows a single location to backup and save data for several systems.

LDAP resources
Before using QoS, you should be familiar with LDAP concepts and directory structures. Please review LDAP basics within the Directory Services (LDAP) topic of the iSeries Information Center.

QoS tree structure
When you want to manage part of your directory, you reference the **Distinguished Name (DN)** or (if you choose) a keyword. You specify the DN when you configure the directory server. DNs usually consist of the name for the entry itself, as well as, the objects (top to bottom) above the entry in the directory. The server can access all objects on the directory that are below the DN. For example, let's say the LDAP server contained the directory structure below:

Figure 12. Sample QoS directory structure



Server1 at the top (dc=server1,dc=chicago,dc=acme,dc=com) is the server on which the directory server resides. The other servers, such as cn=QoS or cn=tcpip policies are where the QoS servers reside. So on cn=server1 the default DN would read `cn=server1,cn=QoS,cn=tcpip`

policies,dc=server1,dc=chicago,dc=acme,dc=com. On cn=server2 the default DN would read cn=server2,cn=QoS,cn=tcpip policies,dc=server1,dc=chicago,dc=acme,dc=com.

When managing your directory, it is important to change the proper server in the DN, such as cn or dc. Be careful when editing the DN, especially since the string is usually too long to be displayed without scrolling. For information on how to configure the directory server within the Quality of service function on iSeries Navigator, see Configure the directory server.

See the related information for QoS page, for some alternative LDAP resources.

Keywords

When you configure your directory server, you will need to determine whether or not to associate keywords to each QoS configuration. The keyword fields are optional and may be ignored. The following information will help explain the keyword concept and why you might want to use them.

In the New Quality of Service Configuration wizard, you will configure a directory server. You will specify whether the server you configure is a primary directory server or a secondary system. The server that you maintain all your QoS policies upon, is known as the primary system.

Keywords are used to identify configurations created by primary systems. Although created on the primary system, keywords are really for the benefit of the secondary system. They allow secondary systems to load and use configurations created by a primary system. The descriptions below will help explain how to use keywords on each system.

Keywords and primary systems
Keywords are associated to QoS configurations created and maintained by a primary system. They are used so secondary systems can identify a configuration created by a primary system.

Keywords and secondary systems
Secondary systems use keywords to search for configurations. The secondary system loads and uses configurations created by a primary system. When you configure a secondary system, you can select specific keywords. Depending on the keyword selected, the secondary system loads any configurations associated with the selected keyword. This allows the secondary system to load multiple configurations created by multiple primary systems.

When you begin to configure the directory server in iSeries Navigator, use the QoS task help for specific instructions.

Integrated services

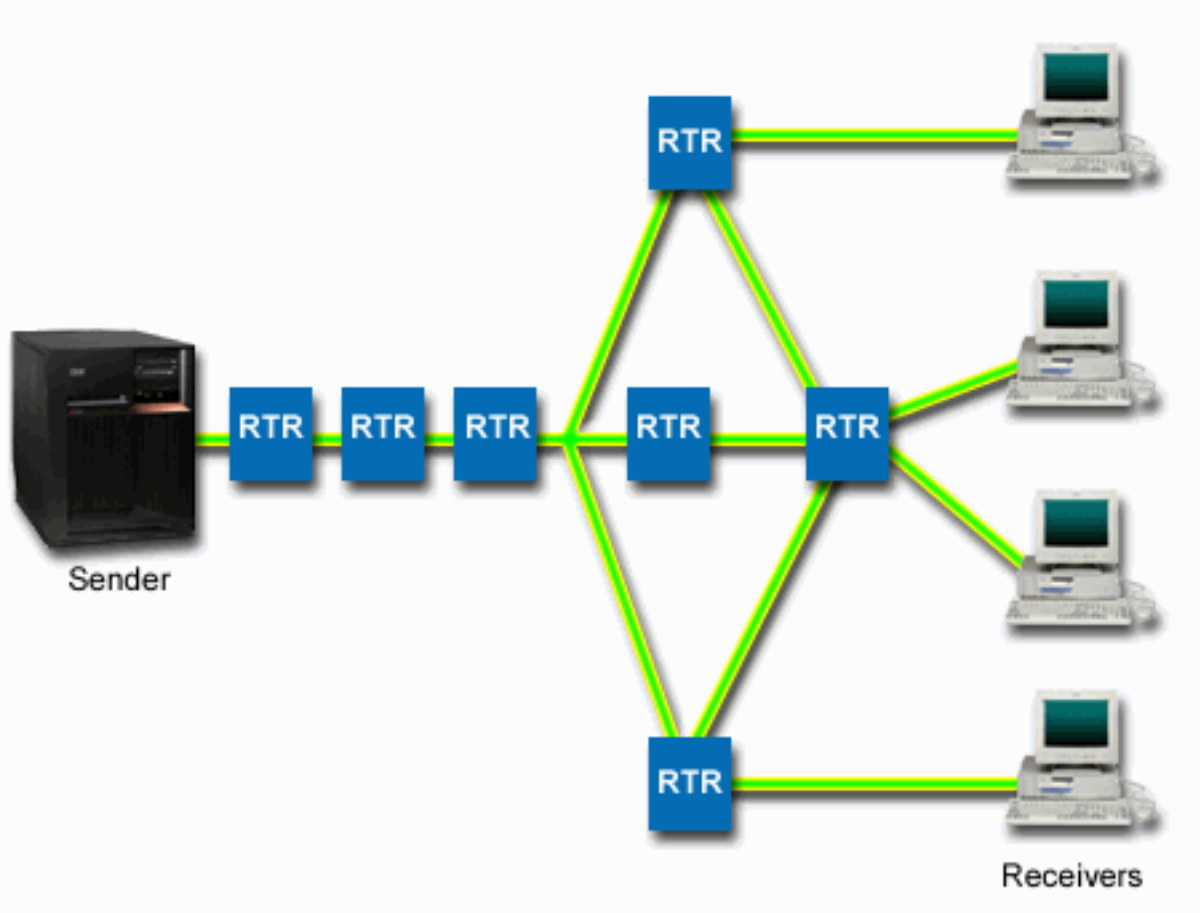
Integrated services deals with traffic delivery time and assigning particular traffic special handling instructions. It is important to be conservative with your integrated services policies because it is still relatively expensive to guarantee data transfer. However, over provisioning your resources can be even more expensive.

Integrated services reserves resources for a particular policy before the data is sent. The routers are signaled before data transfer and the network actually agrees to and manages (end-to-end) data transfer based on a policy. A **policy** is a set of rules that designate an action. It is basically an admission control

list. The bandwidth request comes in a reservation from the client. If all the routers in the path agree to the requirements coming from the requesting client, the request gets to the server and intserv policy. If the request falls within the limits defined by the policy, the QoS server grants permission for the RSVP connection and will then set aside the bandwidth for the application. The reservation is performed using the Resource Reservation Protocol (RSVP) and RAPI and/or qtoq QoS sockets APIs. See RSVP protocol and QoS APIs for more information.

Every node that your traffic travels through must have the ability to use the RSVP protocol. The routers provide quality of service through the following traffic control functions: packet scheduler, packet classifier, and admission control. The ability to carry out this traffic control is often referred to as RSVP-enabled. As a result, the most important part of implementing integrated services policies is being able to control and predict the resources in your network. To get predictable results, every node in the network should be RSVP-enabled. For example, your traffic is routed based on resources, not on which paths have RSVP-aware routers. Crossing routers that are not RSVP-aware may cause unpredictable performance problems. The connection is still made, but the performance that the application requests is not guaranteed by that router. The following figure shows how the integrated service function logically works.

Figure 13. RSVP path between client and server.



The RSVP-enabled application on the server detects a connection request from the client. In response, the server's application issues a PATH command to the client. This command is issued using the RAPI APIs or qtoq QoS sockets APIs and contains router IP address information. A PATH command contains information about the available resources on the server and the routers along the path, as well as, route information

between the server and the client. The RSVP-enabled application on the client then sends a RESV command back along the network path to signal the server that the network resources have been allocated. This command makes the reservation, based on the router information from the PATH command. The server and all routers along the path reserve the resources for the RSVP connection. When the server receives the RESV command, the application starts transmitting data to the client. The data is transmitted along the same route as the reservation. Again, this shows how important the routers' abilities to carry out this reservation are to the success of your policies.

Integrated services is not meant for short term RSVP connections, like HTTP. Of course this is at your discretion. Only you can decide what is best for your network. Consider what areas and applications are having performance problems and need quality of service. Applications used in an integrated services policy must be able to use the RSVP protocol. Currently, your server does not have any RSVP-enabled applications, so you will need to write the application to use RSVP. See the RSVP section for more detail.

As packets arrive and attempt to leave your network, your server determines whether or not it has the resources to send the packet. This acceptance is determined by the amount of space in the token bucket. You manually set the number of bits to allow into your token bucket, any bandwidth limits, token rate limits, and the maximum number of connections your server should allow. These values are referred to as performance limits. If the incoming packets will cause the bucket to exceed its limit, the packets are considered non-conformant. Your server can handle non-conformant traffic in a few different ways. It can either delay, shape, retransmit, or drop the packets. If the packets will remain within the server's limits, the packets conform and are sent out. In integrated services, each connection is granted its own token bucket. In differentiated services, the whole subnet or range of clients share a token bucket.

Traffic control functions

Traffic control function only applies to integrated service policies. To get predictable results, you need to have RSVP-enabled hardware along the traffic's path. Routers must have certain traffic control functions in order to use the RSVP protocol. This is often referred to as being RSVP-enabled or QoS-enabled. Remember that your server's role is either a client or a server. It can not be used as a router at this time.

Traffic control functions include the following:

Packet scheduler
The packet scheduler manages the packet forwarding based on the information in the IP header. The packet scheduler ensures that the packet delivery corresponds to the parameters you set in your policy. The scheduler is implemented at the point where packets are queued.

Packet classifier
The packet classifier identifies which packets of an IP flow will receive a certain level of service based, again, on the IP header information. Each incoming packet is mapped by the classifier into a specific class. All the packets that are classified in the same class receive the same treatment. This service level is based upon the information you provided in your policy.

Admission control
The admission control contains the decision algorithm that a router uses to determine if there are enough routing resources to accept the requested QoS for a new flow. If there are not enough resources, the new flow is rejected. If the flow is accepted, the router assigns the packet classifier and scheduler to reserve the requested QoS. Admission control occurs in each router along the reservation path.

This is not an all-inclusive discussion on classifiers and schedulers. To locate alternative sources, please review the related information for QoS page.

Integrated service types

There are two integrated service types: controlled load and guaranteed.

Controlled Load

Controlled load service supports applications that are highly sensitive to congested networks, such as real time applications. Applications must also be tolerant to small amounts of loss and delay. If an application uses the controlled load service, its performance will not suffer as network load increases. Traffic will be provided with service resembling normal traffic in a network under light conditions.

Routers must ensure that controlled load service receives adequate bandwidth and packet processing resources. To do this, they must be QoS enabled with support for Integrated services. You will need to check the router's specifications to see if they provide quality of service through a traffic control function. Traffic control consists of the following components: packet scheduler, packet classifier, and admission control.

Guaranteed service
Guaranteed service assures that packets will arrive within a designated delivery time. Applications that need guaranteed service include video and audio broadcasting systems that use streaming technologies. Guaranteed service controls the maximum queuing delay, so that packets will not be delayed over a designated amount of time. Every router along the packet's path must provide RSVP capabilities to assure delivery. When you assign the token bucket limits and bandwidth limits, you are defining your guaranteed service.

Token bucket and bandwidth limits

Token bucket limits and bandwidth limits are together known as performance limits. These performance limits help guarantee packet delivery in outbound bandwidth policies, both integrated and differentiated service.

Token bucket size
The token bucket size determines the buffer capacity, which holds bursts of data. Burst data is information that an application gives the server to send out, at a faster rate than it can exit. If an application quickly sends enough burst data to your server, the buffer fills up. If the application sends information slower than it can exit the server, the buffer empties. When data is leaving the server as fast as it is entering the server, then the token bucket size remains unchanged. Once the buffer is filled, QoS treats additional data packets as out-of-profile. In this policy you can determine how QoS handles out-of-profile traffic.

Token rate limit
The rate (bandwidth) limit specifies the long term data rate or the number of bits per second allowed into a network. Any client requesting RSVP from the server will ask for a specific amount of bandwidth (flow limit). The QoS policy looks at the requested bandwidth and compares it with the rate and flow limits for this policy. If the request would cause the server to exceed its limits, the server denies the request. The token rate limit is only used for admission control within integrated service policies. It is measured in Kb/s. This value can vary between 10 Kb/s to 1Gb/s.

The average rate limit or bandwidth limit must be less than the peak rate limit or peak bandwidth limit, so you don't exhaust the entire interface. For example, imagine you have a modem using 36 Kb/s or smaller, you'll need to set your average rate limit such, that the whole interface will not be utilized.

Hint: To determine what limits to set, you may want to run the monitor. Create a policy with an aggregate token rate limit large enough to collect most data traffic on your network. Then start data collection on this policy. See the Monitor current network statistics example for one way to collect the total rates your application and network currently use. Using these results, you can reduce the limits appropriately.

See the differentiated classes of service and integrated services topics for more information.

Integrated service using differentiated service markings

This policy is most often used when you have a mixed environment. A mixed environment occurs when an integrated service reservation travels through different routers which don't support integrated service reservations, but do support differentiated services. Since your traffic passes through different domains, service level agreements, and equipment capabilities, you may not always get the service you intend.

To help alleviate this potential problem, you can attach a differentiated services marking to your integrated services policy. In the event that a policy crosses a router that cannot use the RSVP protocol, your policy will still maintain some priority. The marking you add is called a per-hop behavior.

No signalling

In addition to using markings, as described above, you can also use the new "no signal" function. The "No Signal" is specified within the integrated service policy. You designate no signal on the **Properties** panel of any integrated service policy.

1. In iSeries Navigator, expand your server —> **Network—> IP Policies**.
2. Right-click **Quality of Service** and select **Configuration**.
3. Expand **Outbound bandwidth policies —> IntServ**.
4. Right-click on the policy name you created above and select **Properties**, the IntServ Properties dialog appears.
5. Select the **Traffic Management** tab to disable or enable signalling. This is also where you edit the schedule, client, applications, and traffic management.

When selected, the "no signal" versions of the APIs will allow you to write an application that causes an RSVP rule to be loaded on the server and only requires the server side application of the TCP/IP conversation to be RSVP-enabled. The RSVP signalling is done automatically on behalf of the client side. This creates the RSVP connection for the application even if the client side is not able to use the RSVP protocol.

See the differentiated classes of service and integrated services topics for more information.

RSVP protocol and QoS APIs

The Resource Reservation Protocol (RSVP), along with the RAPI APIs or qtoq QoS sockets APIs perform your integrated service reservation. Every node that your traffic travels through must have the ability to use the RSVP protocol. The ability to carry out integrated services policies is often referred to as RSVP-enabled. For more information on what router functions are needed to use the RSVP protocol, see Traffic control functions.

The RSVP protocol is used to create an RSVP reservation in all the network nodes along your traffic's pathway. It maintains this reservation long enough to provide your policies requested services. The reservation defines the handling and bandwidth that the data in this conversation will require. The network nodes each agree to provide the data handling defined in the reservation.

RSVP is a simple protocol in that reservations are only made in one direction (from the receiver). For more complex connections, such as audio and video conferences, each sender is also a receiver. In this case, you must set up two RSVP sessions for each side.

In addition to RSVP-enabled routers, you need to have RSVP-enabled applications to use integrated services. Since the iSeries server does not have any RSVP-enabled applications at this time, you will need

to write the applications using the RAPI API or the qtoq QoS Sockets APIs. This will enable the applications to use the RSVP protocol. If you want an in-depth explanation, there are many sources that explain these models, their operation, and messaging. You need a thorough understanding of the RSVP protocol and the contents of Internet RFC 2205.

qtoq Sockets APIs
You can now use the qtoq QoS sockets APIs to simplify the work required to use the RSVP protocol on the iSeries system. The qtoq sockets APIs call the RAPI APIs and perform some of the more complex tasks. The qtoq sockets APIs are not as flexible as the RAPI APIs, but provide the same function with less effort. The ″No Signal″ versions of the APIs allow you to write the following:

- An application that will load an RSVP rule on the server.
- An application that only requires the server side application (of the TCP/IP conversation) to be RSVP-enabled.

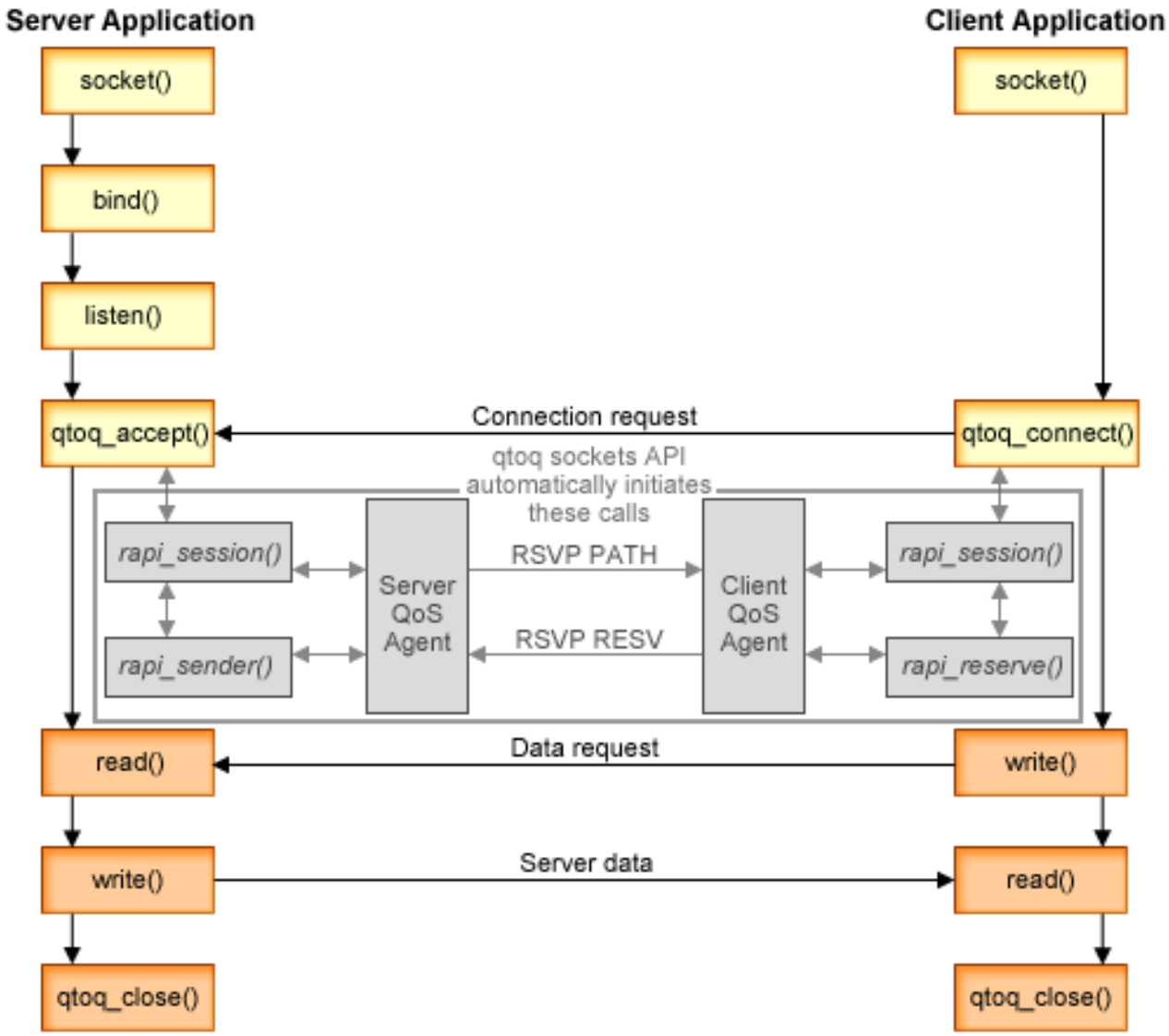
The RSVP signalling is done automatically on behalf of the client side.

See the QoS API Connection oriented functional flow page, or the QoS API Connectionless functional flow page for typical QoS API flow for an application/protocol using connection oriented or connectionless qtoq QoS sockets.

QoS API Connection Oriented Functional Flow

The following figure illustrates the client/server relationship of the QoS enabled API qtoq sockets functions for a connection-oriented protocol, such as Transmission Control Protocol (TCP).

When the QoS enabled API functions are called for a connection oriented flow requesting that RSVP be initiated, additional functions are initiated. These functions cause the QoS agents on the client and server to set up the RSVP protocol for the data flow between the client and the server.



qtoq flow of events: The following sequence of socket calls provide a description of the graphic. It also describes the relationship between the server and client application in a connection-oriented design. These are modifications of the basic Sockets APIs.

Server side

qtoq_accept() for a rule marked "No Signaling"

1. The application calls the socket() function to get a socket descriptor.
2. The application calls listen() to specify what connections it will wait for.
3. The application calls qtoq_accept() to wait for a connection request from the client.
4. The API calls the rapi_session() API and, if successful, a QoS session ID will be assigned.
5. The API calls standard accept() function to wait for a client connection request.
6. When the connection request is received admission control is performed on the requested rule. The rule is sent to the TCP/IP stack, if valid, it returns to the calling application with the results and the session ID.

7. The applications for the server and the client perform the desired data transfers.
8. The application will call the qtoq_close() function to close the socket and unload the rule.
9. The QoS server will delete the rule from the QoS manager, delete the QoS session, and perform whatever other cleanup is needed.

qtoq_accept() with normal RSVP signalling

1. The application calls the socket() function to get a socket descriptor.
2. The application calls listen() to specify what connections it will wait for.
3. The application calls qtoq_accept() to wait for a connection request from the client.
4. When a connection request comes in the rapi_session() API will be called to create a session with the QoS server for this connection and get the QoS session ID which will be returned to the caller.
5. The rapi_sender() API will be called to initiate a PATH message from the QoS server and inform the QoS server that it must expect a RESV message from the client.
6. The rapi_getfd() API is called to get the descriptor that the applications use to wait for QoS event messages.
7. The accept descriptor and the QoS descriptor are returned to the application.
8. The QoS server waits for the RESV message to be received. When the message is received it will load the appropriate rule with the QoS manager and send a message to the application if the application requested notification on the qtoq_accept() API call.
9. The QoS server continues to provide refreshes for the established session.
10. The application calls qtoq_close() when the connection is completed.
11. The QoS server will delete the rule from the QoS manager, delete the QoS session, and perform whatever cleanup is needed.

Client side

qtoq_connect() with normal RSVP signalling

1. The application calls the socket() function to get a socket descriptor.
2. The application calls qtoq_connect() function to inform the server application that it would like to make the connection.
3. The qtoq_connect() function calls the rapi_session() API to create a session with the QoS server for this connection.
4. The QoS server will be primed to wait for the PATH command from the requested connection.
5. The rapi_getfd() API is called to get the QoS descriptor that the applications use to wait for QoS messages.
6. The connect() function is called. The results of the connect() and the QoS descriptor are returned to the application.
7. The QoS server waits for the PATH message to be received. When the message is received it will respond with a RESV message to the QoS server on the applications server machine.
8. If the application requested notification, the QoS server will send the notification to the application via the QoS descriptor.
9. The QoS server continues to provide refreshes for the established session.
10. The application calls qtoq_close() when the connection is complete.
11. The QoS server will close the QoS session and perform whatever cleanup is necessary.

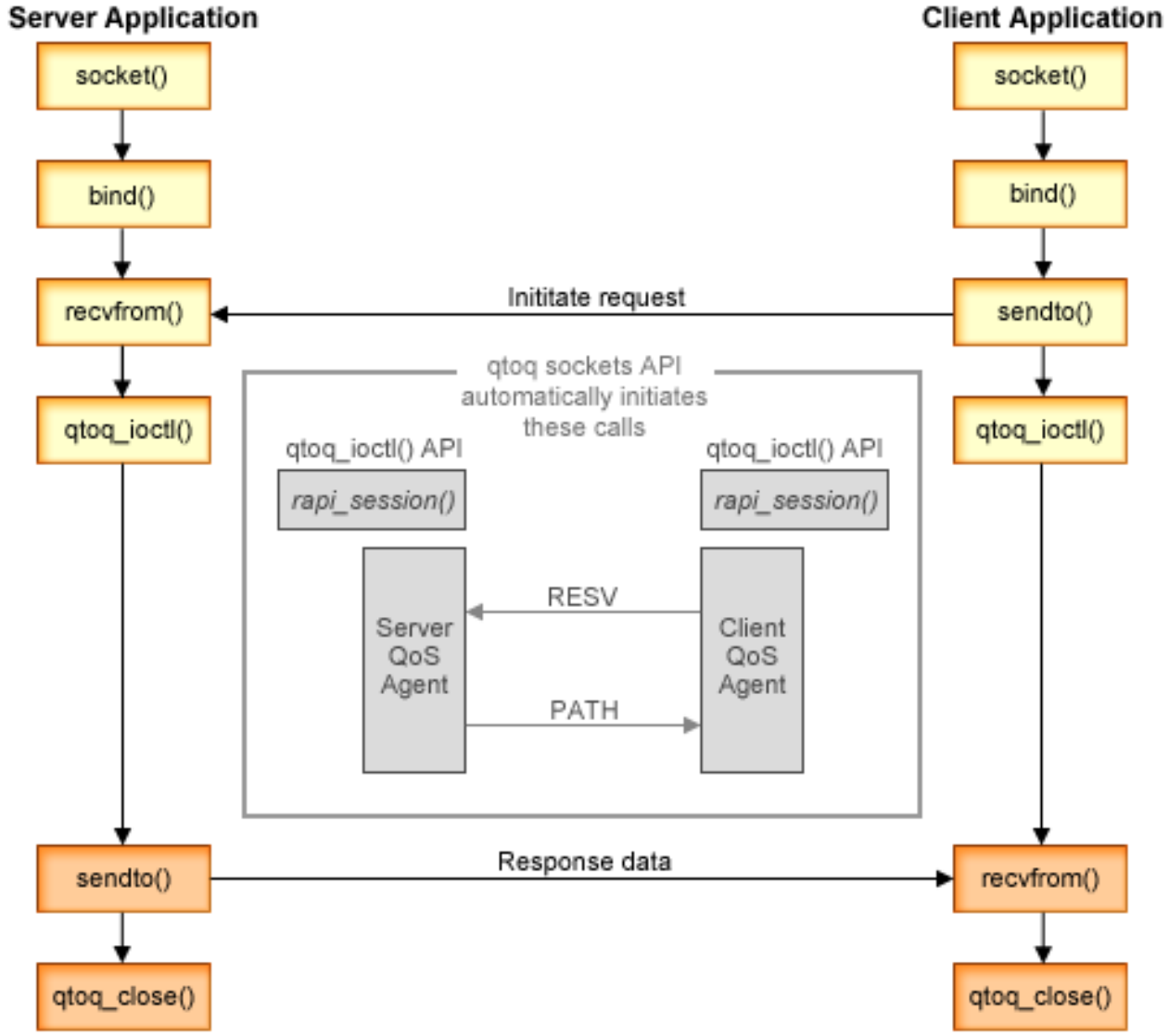
qtoq_connect() for a rule marked "No Signaling"

This request is not valid for the client side, since no response is required from the client in this case.

QoS API Connectionless Functional Flow

These server and client examples illustrate qtoq QoS socket APIs written for a connectionless flow.

When the QoS enabled API functions are called for a connectionless flow requesting that RSVP be initiated, additional functions are initiated. These functions cause the QoS agents on the client and server to set up the RSVP protocol for data flow between the client and server.



qtoq flow of events: The following sequence of socket calls provide a description of the graphic. It also describes the relationship between the server and client application in a connectionless design. These are modifications of the basic Sockets APIs.

Server side

qtoq_ioctl() for a rule marked ″No Signaling″

1. Sends a message to the QoS server asking it to perform admission control on the requested rule.
2. If the rule is acceptable, it calls a function that sends a message to the QoS server requesting that the rule be loaded.
3. Returns status to the caller indicating success or failure of the request.
4. When the application has completed using the connection, it calls the qtoq_close() function to close the connection.
5. The QoS server will delete the rule from the QoS manager, delete the QoS session and perform whatever other cleanup is needed.

qtoq_ioctl() with normal RSVP signalling

1. Sends message to the QoS server requesting admission control for the requested connection.
2. Calls rapi_session() to request a session be set up for the rule and get the QoS session ID to be returned to the caller.
3. Calls rapi_sender() to initiate a PATH message back to the client.
4. Calls rapi_getfd() to get file descriptor in order to wait for QoS events.
5. Returns descriptor select(), QoS session ID and status to the caller.
6. QoS server loads rule when the RESV message is received.
7. Application issues a qtoq_close() when the connection is completed.
8. The QoS server will delete the rule from the QoS manager, delete the QoS session, and perform whatever cleanup is needed.

Client side

qtoq_ioctl() with normal RSVP signalling

1. Calls rapi_session() to request a session be set up for the connection. The rapi_session() function requests admission control for the connection. The connection will only be rejected on the client side if there is a configured rule for the client and it is not active at this time. This function returns the QoS session ID that is passed back to the application.
2. Calls rapi_getfd() to get file descriptor in order to wait for QoS events.
3. The qtoq_ioctl() returns back to the caller with the wait on descriptor and session ID.
4. The QoS server waits for the PATH message to be received. When the path message is received it will respond with the RESV message and then signal the application that the event has occurred via the session descriptor.
5. The QoS server continues to provide refreshes for the established session.
6. The client code calls qtoq_close() when the connection is completed.

qtoq_ioctl() for a rule marked ″No Signaling″

This request is not valid for the client side, since no response is required from the client in this case.

Plan for QoS

The most important step to implementing quality of service is planning. To receive expected results, you must review your network equipment and monitor network traffic. The QoS planning advisor leads you through the basic questions you need to ask yourself during the planning phase. In addition to the advisor, consider these subtopics before implementing QoS.

Authority requirements
Lists all the authorities you need to configure QoS and a directory server successfully.

System requirements
Lists all the requirements you need to operate QoS successfully.

Order QoS policies
The order your policies appear in the file is the order they are processed. This only applies for differentiated service policies and connection rate policies.

Service level agreements
Service level agreements are an important part of QoS. You must understand and set up a SLA with your network provider as part of your QoS planning.

Network hardware and software
Quality of service is only as good as its weakest link. The capabilities of your internal equipment and other equipment outside your network have enormous effects on QoS results.

Network performance
QoS is all about network performance. The main reason you are considering QoS is probably because you are already experiencing network congestion and packet loss. Before you implement any policies, you may want to use the QoS monitor to verify your IP traffic's current performance levels. These results will help you determine where congestion is occurring. See the Monitor server transactions topic under Troubleshooting.

QoS planning advisor
Consider these basic questions before you implement quality of service. You receive a planning worksheet with suggested policies based on your applications' abilities.

Authority requirements

Quality of service policies may contain sensitive information about your network. Therefore QoS administrative authority should only be granted when necessary. The following authorities will be required before you can configure QoS policies or LDAP directory servers. Since QoS policies are stored on an LDAP directory server, both authorities are required.

Grant authorities needed to manage the directory server
The QoS administrator will need the following authority: `*ALLOBJ` authority and `*IOSYSCFG`. See Configure directory server for alternative authorities.

Grant authority to start the TCP/IP server.
To grant object authority to the STRTCPSVR and ENDTCPSVR commands, follow these steps:

1. **STRTCPSVR**: At the command line, type GRTOBJAUT OBJ (QSYS/STRTCPSVR) OBJTYPE (*CMD) USER (ADMINPROFILE) AUT (*USE), substituting the name of your administrator's profile for ADMINPROFILE, and press **Enter**.
2. **ENDTCPSVR**: At the command line, type GRTOBJAUT OBJ (QSYS/ENDTCPSVR) OBJTYPE (*CMD) USER (ADMINPROFILE) AUT (*USE), substituting the name of your administrator's profile for ADMINPROFILE, and press **Enter**.

Grant all object access and system configuration authorities.
It is recommended that users who will be configuring QoS have security officer access. To grant all object access and system configuration authorities, follow these steps:

1. In iSeries Navigator, expand your server —> **Users and Groups**.
2. Double-click **All users**.
3. Right-click the administrator's user profile and select **Properties**.
4. On the Properties dialog, click **Capabilities**.
5. On the Capabilities page, select **All object access and System configuration**.
6. Click **OK** to close the Capabilities page.
7. Click **OK** to close the Properties dialog.

System requirements

Quality of service (QoS) is an integrated part of the operating system. Before you proceed with configuring and starting QoS, you must have at least the Version 5 Release 1 OS/400(R). In addition, you must complete these requirements:

1. Install TCP/IP Connectivity Utilities (57*xx*-TC1).
2. Install iSeries Navigator on your PC. Make sure to install the Networking section during the Client Access install. Quality of service is located under IP Policies within Networking.

Note: If you need more information about TCP/IP, networking, or IP addresses, refer to TCP/IP Tutorial and Technical Overview and V4 TCP/IP for AS/400(R): More Cool Things Than Ever in Related information for QoS.

Order QoS policies

Whenever you have two differentiated service policies that overlap or two connection rate policies that overlap, the physical order of your policies in iSeries Navigator is important. Overlapping policies are two policies that use the same client, application, schedules, or protocols. The policies on the iSeries Navigator screen are in an ordered list. Policy precedence depends on the order of the policies in this list. If you want one policy to take priority over another, the higher priority policy must appear in the list first.

To determine if a policy overlaps with another policy, follow these instructions:

1. In iSeries Navigator, expand your server —> **Network —> IP Policies**.
2. Right-click **Quality of Service**.
3. Select **Configuration**.
4. Select the specific Policies folder.
5. Right-click the name of the policy that has associated overlapping policies. Overlapping policies have an icon in front of them to indicate the overlap.
6. Select **Show Overlap**. The overlap panel will appear.

To change policy order on the screen, use the following steps:

- Highlight the policy and use the up and down arrows on the screen to change policy order.
- Right-click the policy name and select **Move up** or **Move down**.
- Update the QoS server. You can use the Update server button on the toolbar or see the QoS task help for more detailed instructions.

Service level agreement

This section is not educational information on service level agreement (SLA) providers, but points out some of the important aspects in your SLA that may affect your quality of service implementation. Your policies and reservations are only as good as the weakest link. This means, if one node anywhere between the client and the server is unable to perform any of the traffic-handling characteristics discussed in the Differentiated services or Integrated services topics, your policies will not be handled as you intended. If your SLA does not allow you enough resources, even the best policies will not help your network's congestion problem.

This also involves agreements across ISPs. Across domains, every ISP must agree to support quality of service requests. Interoperability might cause some challenges.

Make sure that you understand the service level that you are actually receiving. Traffic conditioning agreements specifically address how traffic is handled, that is dropped, marked, shaped, or re-transmitted. The key reasons to provide quality of service involve controlling latency, jitter, bandwidth, packet loss, availability, and throughput. Your service agreements must be able to give your policies what they request. Verify that you are receiving the amount of service you need. If not, you may waste your resources. For example, if you ask to reserve 500kbps for IP telephony, but your application only needs 20kbps you may pay extra without receiving any notice from your ISP.

Network hardware and software

The capabilities of your internal equipment and other equipment outside your network have enormous effects on QoS results.

Applications
Integrated service policies require RSVP-enabled applications. Since the iSeries applications are not presently RSVP-enabled, you must enable them to use the RSVP protocol. To enable your applications, you need to write special programs using the Resource Reservation Setup Protocol (RAPI) APIs or qtoq QoS sockets APIs. These programs will allow your applications to use RSVP. See RSVP protocol and QoS APIs for more information.

Network nodes
The routers, switches, and even your own servers must have the capability to use quality of service. To use differentiated services policies, your equipment must be differentiated services-enabled. This means that the network node must be able to classify, meter, mark, shape, and drop IP packets. For more detailed information about traffic conditioners (classify, meter, mark, shape, and drop) see the Traffic conditioners topic.

To use integrated services policies, your equipment must be RSVP-enabled. This means that the network nodes must also be able to support the RSVP protocol. For more detailed information about the RSVP protocol, see the RSVP topic.

Configure QoS

You create your QoS policies using wizards within iSeries Navigator. The wizards do an excellent job of leading your through configuration.

After you configure your policies, you can use the configuration objects in iSeries Navigator to edit your policy configuration. The configuration objects are the different pieces or parts that make up a policy. When you open quality of service in iSeries Navigator, there are folders labeled clients, applications, schedules, policies, classes of service, per-hop behaviors, and URIs. These objects allow you to create a policy. For more detailed information about the objects, you can see the Quality of service overview help in iSeries Navigator.

Configure directory server
Use this for information about how to configure the directory server within QoS.

Configure QoS using wizards
Use this for instructions on how to access the QoS wizards.

Enable QoS
Before your policies can take effect, you must enable them. If you used the wizards, the server will automatically enable the policies for you. If you changed a policy using the configuration objects, you will need to dynamically update the server before the policies will become active. Before you enable, be sure to look for overlapping policies that may cause problems. See Order QoS policies for more information.

Configure directory server

QoS policy configurations are now stored on an LDAP directory server. This makes your QoS solution easier to manage. Instead of configuring QoS policies on all of your servers, you can store the configuration data on one local directory server for many systems to share. When you first configure quality of service on your server, an Initial Configuration wizard appears. This wizard will prompt you to configure a directory server.

To configure the directory server you will need to decide or know the following information:

- Directory server name
- Determine a distinguished name (DN) to reference the QoS policies
- Determine whether or not to use SSL security with your LDAP directory server
- Determine whether or not to use keywords to improve the search for your policies on the directory server.

Note: Currently, Kerberos cannot be configured as the authentication method the QoS server will use to access the directory.

To administer the LDAP directory server, you must have one of the following authority sets:

- *ALLOBJ authority and *IOSYSCFG authority
- *JOBCTL authority and object authority to the End TCP/IP (ENDTCP), Start TCP/IP (STRTCP), Start TCP/IP Server (STRTCPSVR), and End TCP/IP Server (ENDTCPSVR) commands.
- *AUDIT authority to configure OS/400(R) security auditing.

If you are using iSeries Navigator, you will already have access to the default QoS Schema. However, if you are using an editor other than iSeries Navigator, you will need to import the LDIF file described below. You can also import this file, if after editing, you want to reload the original, default file.

QoS Schema
A set of rules, called a schema, exist to specify what types of LDAP objects are valid to the QoS server. The schema on V5R2 iSeries servers contains the necessary rules for QoS. If, however, the LDAP server used is not an iSeries server, these rules must be imported to the LDAP server. This is done with an LDIF (LDAP Data Interchange Format) file. Use the iSeries LDAP web page

to download the LDIF file. You will find this file under **Categories —> TCP/IP Policies** on the left hand pane.

Edit LDIF file
You can use the IBM(R) SecureWay(R) Directory Management Tool (DMT) to edit the schema files for your LDAP server. You may also FTP the setup.exe file for the DMT to your PC. The setup.exe file should be located on your server at `/qibm/proddata/os400/dirsrv/UserTools/Windows`. The original QoS schema can be obtained from the iSeries LDAP web page. See LDAP concepts for a sample QoS schema. The schema file is located on your server at `/QIBM/UserData/OS400/DirSrv`.

Configure QoS with wizards

To configure quality of service policies, you must use the QoS wizards located in iSeries Navigator. Here is a list of the wizards and their function:

Initial Configuration wizard
This wizard allows you to set up system specific configuration and directory server information.

New IntServ Policy wizard
The new IntServ Policy wizard allows you to create an integrated service policy. This policy admits or denies an RSVP request, which indirectly controls server bandwidth. The policy performance limits (which you set) decide if the server can handle the requested bandwidth coming from the client's RSVP application. You will need RSVP ready routers and applications to implement the integrated service policies created in this wizard.

> **Note:** Before you set up an integrated service policy you must write your own applications to use the RSVP protocol. For more information, see RSVP protocol and the QoS APIs.

New DiffServ policy wizard
This wizard allows you to differentiate and assign priority to TCP/IP traffic. You will be able to differentiate traffic by creating policies. Within a policy, you can assign priority to applications and ports, and specify when this policy should be active.

New DiffServ classes of service wizard
Use the differentiated class of service wizard to set packet markings used by routers and switches within networks. It also assigns performance limits to the traffic leaving your network. You use classes of service with a DiffServ policy.

New Connection rate wizard
Use the Inbound connection rate wizard to restrict connections being made to your server. You can restrict access by TCP/IP address, by application, or by local interface. This allows a system administrator to control access to your server from specific clients or to server applications or interfaces. In addition, you may enhance server performance.

New URI wizard
Use the Inbound URI wizard to restrict connections being made to your server. You can restrict access by URI, by application, or by the local interface on your iSeries server. This allows a system administrator to control access to specific URIs, applications, or interfaces on your server. In addition, you may enhance server performance.

> **Note:** Before you set up URI request rate policies, you must perform the following steps:
> 1. `WRKHTTPCFG` - modify your Apache Web server instance. Enable a port via the Listen directive with the Fast Response Cache Accelerator (FRCA) option.
> 2. `STRTCPSVR SERVER(*HTTP) HTTPSRV`(name of instance).

3. Create or modify a URI policy using QoS within iSeries Navigator. Make sure the application port defined in the URI policy matches the FRCA ″Listen directive″ defined in the Apache Web Server instance.
4. `STRTCPSVR SERVER(*QOS).`

The application port assigned in the new URI policies must match the 'Listen' directive enabled for FRCA in the Apache Web Server configuration. If the port values do not match, the QoS URI policy will not function as expected. For a description of URI request rate policies, see Connection request rate and URI request rate.

Once you decide which type of policy to create, you can configure the policy in the appropriate wizard listed above. See Access the QoS wizards in iSeries Navigator to start to configure a policy.

Access the QoS wizards within iSeries Navigator

To access the QoS wizards and create a new policy, follow these steps:

1. In iSeries Navigator, expand your server —> **Network—> IP Policies**.
2. Right-click **Quality of Service** and select **Configuration**.
   Note: The Initial Configuration wizard appears in the following circumstances:
   - You are upgrading your server to a new release. You will need to configure the directory server in which to store information. No data is lost during this conversion.
   - This is the first time you are using the QoS graphical user interface (GUI) on this system.
   - You want to manually remove any previous configuration information and start over. This only occurs if the QoS interface is already open.
3. Complete the **Initial Configuration wizard**. If the Initial Configuration wizard does not appear, skip to step 4.
4. Select **Policies**. Right-click on either **IntServ**, **DiffServ**, **Connection rate**, or **Server request —> URI**.
5. Select **New Policy**.

Manage QoS

Once you have your QoS policies active and running, you will probably need to make updates. You can manage your policies by doing the following:

Access QoS task help in iSeries Navigator
You probably noticed that this topic refers to the QoS task help in iSeries Navigator quite often. If you are not sure how to get there, review these instructions.

Back up QoS policies
You can back up your policies to protect yourself against losing files.

Copy an existing policy
You can copy an existing policy that may be similar to the policy you want to create.

Dynamically update policies
You can dynamically update policies while your server is running. Use *Update the QoS server* in the QoS task help of iSeries Navigator for step-by-step instructions.

Edit QoS policies
You can change parameters in your existing policies.

Edit QoS configuration properties
You can change the properties of your quality of service configuration. These properties include settings for the directory server configuration, journaling, and automatically starting the server. Use *Edit QoS properties* in the QoS task help of iSeries Navigator for step-by-step instructions.

Enable QoS policies
Before your policies can take effect, you must enable them. Remember to manually check for possible errors before you enable the policies. For example, be sure your policies are in the correct order. If you want more information about policy order, see Order QoS policies. Otherwise, use*Enable QoS policies* in the QoS task help of iSeries Navigator for step-by-step instructions.

Monitor QoS policies
As you manage your policies, you may want to analyze the QoS monitor to verify that the policies are working as you intend.

View QoS policies
By viewing overlapping policies, you can determine where you may have different results than what you expect. You can check for any visible overlaps between policies that may cause problems. You will want to view these overlaps not only before activating and testing, but also before printing and backing up. This is a useful way to minimize or remove the errors before testing. To view overlapping policies, see Order QoS policies.

Access QoS help in iSeries Navigator

To access the quality of service help, you must use iSeries Navigator:

1. In iSeries Navigator, expand your server —> **Network—> IP Policies**.
2. Right-click **Quality of Service** and select **Configuration**.
3. Select **Help —> Help topics** from the menu bar. The task help window appears on your screen.

Back up QoS policies

Backing up your configuration files is always a good idea. Your policies are stored both locally and in a directory server. You should specifically backup the following integrated file system directory: `QIBM/UserData/OS400/QOS/ETC, QIBM/UserData/OS400/QOS/TEMP, and QIBM/UserData/OS400/QOS/USR`. You should also backup your directory server publishing agent for the QoS server. The publishing agent contains the directory server name, the distinguished name (DN) for the QoS server, port used to access the directory server, and authentication information. In the event of a loss, your backups can save you the time and work it would take to recreate your policies from scratch. These are general tips you can use to ensure that you have an easy way to replace lost files:

1. **Use integrated file systems backup and recovery programs**
   Use the link to the Backup and Recovery book seen below.
2. **Print out the policies**
   You can store the printouts wherever they are most likely to be secure and re-enter the information as necessary.
3. **Copy the information to a disk**
   Copying has an advantage over printouts: rather than reentering manually, the information exists electronically. It provides you a straightforward method for transporting information from one on-line source to another.

   Note: Your iSeries server copies information to the system disk, not to a floppy disk. The rules files are in `QIBM/UserData/OS400/QOS/ETC` as well as, within the distinguished name in the directory server you configured, not on a PC. You may want to use a disk protection method as a backup means for protecting the data that is stored on the system disk.

When using an iSeries server, you must plan a backup and recovery strategy. Review Backup and Recovery

for more detailed information.

Copy an existing policy

You may find that you have a few policies that are very similar to one another. Rather than create all of them from scratch, you can make copies of the original policy and then edit the sections of the policy which differ from the original policy. In iSeries Navigator, this QoS function is called *New based on*. You must use iSeries Navigator to access the QoS dialog that enables you to proceed with copying policies.

To create a copy of an existing policy, follow the steps in **Create a new policy based on an existing policy** within the iSeries Navigator help.

Before your policies can take effect, you must enable them by starting the QoS server or performing a dynamic server update. Before you enable, be sure to look for overlapping policies that may cause problems. See Order QoS policies for more information.

Monitor QoS

You can use the monitor to analyze your IP traffic through the server. This helps to determine where congestion is occurring within your network. Not only is this useful during QoS planning, but it can also be helpful as a troubleshooting tool. The QoS monitor can help you continue to monitor your network so you can adjust your policies as needed.

To run the QoS monitor, use the instructions in the iSeries Navigator QoS help.

Note: If QoS data collection is turned on and you plan to make changes to the QoS configuration, then you must perform the following steps to ensure the monitor collects accurate data.

1. Stop the QoS Data Collection.
2. Make Configuration Changes.
3. Restart/Update QoS Server.
4. Start the QoS Data Collection.

Monitor output

The output information you receive depends on the type of policy you are monitoring. Remember the types of policies: DiffServ, IntServ (Controlled Load), IntServ (Guaranteed), Connection rate, and URI. The fields to evaluate depend on the policy type. The most interesting values are the values that show a measurement. The following fields are measured rather than a given definition: accepted requests, active connections, connections services, connection rates, dropped requests, in-profile packets, in-profile bits, non-conformant bits, out-of-profile bits, total bits, total packets, and total requests.

By reading information from the measured fields above, you can form a good picture of how your network traffic is conforming to your policies. Use the descriptions below for more detailed information on the monitor output field for each policy type. For your reference, see any of the QoS scenarios for a sample of how to use the monitor along with the QoS policies.

- Differentiated service policies (See 43)
- Integrated service (controlled load) policies (See 44)
- Integrated service (guaranteed) policies (See 45)
- URI policies (See 46)
- Connection rate policies (See 45)

Differentiated service policies

Field	Description
Policy name	The name you assigned to this policy.
Protocol	UDP, TCP, ALL
Average token rate limit	The average token rate allowed by this policy in each router and server along the flow path.
Token depth limit	The maximum token buffer size allowed by this policy in each router and server along the flow path.
Peak token rate limit	The maximum rate allowed by this connection.
Packets in-profile	The number of transmitted IP packets that fit within this policy's parameters.
Bits in-profile	The number of transmitted bits that fit within this policy's parameters.
Bits out-of-profile	The number of transmitted bits that exceed the policy's parameters.
Bits rate	The measured number of bits permitted by this connection.
Active connections	The total number of active connections.
Traffic profile	The type of packet conditioning used on out-of-profile packets. Format may include: • Re-marking • Shaping • Dropping

Bits total	The number of transmitted bits used by this policy from the time it was started to the time of the monitor collection.
Codepoint in-profile	If the packet is remarked with a new codepoint, this is the codepoint which IP packets will use if they fit within this policy's parameters.
Codepoint out-of-profile	If the packet is remarked with a new codepoint, this is the codepoint which the IP packets will use if they exceed the policy's parameters.
Destination address range	The address range which determines the packets' (controlled by this policy)destination point.
Packet total	The number of packets transmitted by this policy from the time the policy started to the time of the monitor collection.
Source port range	The source port range which determines which applications are controlled by this policy.

Integrated service (controlled load) policies

Field	Description
Policy name	The name you assigned to this policy.
Protocol	UDP or TCP
Destination address	The address range which determines the packets' (controlled by this policy)destination point.
Average token rate limit	The average token rate allowed by this policy in each router and server along the connection path.
Token depth limit	The maximum token buffer size allowed by this policy in each router and server along the connection path.
Peak token rate limit	The maximum rate allowed by this connection.
Packet total	The number of packets transmitted by this policy from the time the policy started to the time of the monitor collection.
Bits non-conformant	The number of transmitted bits that exceed the policy's parameters.
Bits total	The number of transmitted bits used by this policy from the time it was started to the time of the monitor collection.
Bit rate	The measured number of bits permitted by this connection.
Bits conformant	The number of transmitted bits that fit within this policy's parameters.
Maximum packet size	The maximum allowed packet size controlled by this policy.
Minimum policed unit	The smallest number of bits that will be removed from the token bucket. For example, if your minimum policed unit is 100 bits, packets under 100 bits will still be removed at 100 bits.
Packets conformant	The number of transmitted IP packets that fit within this policy's parameters.

Source port range	The source port range which determines which applications are controlled by this policy.

Integrated service (guaranteed) policies

Field	Description
Policy name	The name you assigned to this policy.
Protocol	UDP or TCP
Destination address	The address range which determines the packets' (controlled by this policy)destination point.
Average token rate limit	The maximum token rate allowed by this policy in each router and server along the connection path.
Token depth limit	The maximum token buffer size allowed by this policy in each router and server along the connection path.
Peak token rate limit	The maximum rate allowed by this connection.
Packet total	The number of packets transmitted by this policy from the time the policy started to the time of the monitor collection.
Bits total	The number of transmitted bits used by this policy from the time it was started to the time of the monitor collection.
Bits non-conformant	The number of transmitted bits that exceed the policy's parameters.
Guaranteed rate	The guaranteed rate in bits per second.
Bits conformant	The number of transmitted bits that fit within this policy's parameters.
Maximum packet size	The maximum allowed packet size controlled by this policy.
Minimum policed units	The smallest number of bits that will be removed from the token bucket. For example, if your minimum policed unit is 100 bits, packets under 100 bits will still be removed at 100 bits.
Packets conformant	The number of transmitted IP packets that fit within this policy's parameters.
Slack term	The difference (in seconds) between the desired delay and the delay obtained.
Source port range	The source port range which determines which applications are controlled by this policy.

Connection rate policies

Field	Description
Policy name	The name you assigned to this policy.
Connection rate	The number of connection requests accepted per second.
Total requests	The total number of connection requests made to this server.
Accepted requests	The total number of connection requests accepted by this server.
Dropped requests	The total number of requests dropped by this server.

Average connection rate limit	The average allowable number of new connection requests admitted per second.
Connection burst limit	The maximum number of new connection requests accepted concurrently .
Peak connection rate limit	The maximum allowable rate at which the server will accept connections from the network
Priority	The priority assigned to each rule loaded in the QoS Manager.
Queue Priority	The priority assigned to incoming connections placed in the listen queue.
Destination port range	The port range or port to which traffic is destined on your server.
Interface address	IP address of system interface being monitored.
Source address range	The IP address range of the clients sending requests to your server.

Server request - URI policies

Field	Description
Policy name	The name you assigned to this policy.
Request rate	The number of requests received per second.
Total Requests	The total number of requests received by the target server.
Accepted requests	The total number of requests accepted.
Dropped requests	The total number of requests dropped.
URI	The identity of the URI being policed.
Average request rate limit	The average allowable number of new requests admitted per second.
Request burst limit	The maximum number of new requests accepted concurrently.
Peak request burst limit	The maximum allowable rate at which the server will accept requests from the network
Queue priority	The priority assigned to incoming connections placed in the listen queue.
Destination port	The port to which traffic is destined on your server.
Interface address	IP address of system interface being monitored.

Troubleshoot QoS

This subtopic provides troubleshooting advice for QoS problems.

Communications trace
Your server provides a communication trace to collect data on a communication line, such as a local area network (LAN) or wide area network (WAN) interface. The average user may not understand the entire contents of the trace data. However, you can use the trace entries to determine whether a data exchange between two points actually took place. For more information, see Communications trace within the TCP/IP Troubleshooting topic.

Enable QoS on the server
The first thing to check if the QoS server does not start, is the value of `IPQOSENB` using the `CHGTCP` command. When you configure your policies for the first time, the Initial configuration wizard automatically enables QoS on the server. If this value has been changed for any reason, the server will not start. From a command line interface, enter `CHGTCPA IPQOSENB(*YES)`.

Journal QoS policies
Your quality of service function includes a journaling feature. You can use journaling to log IP policies added, removed, or modified on your server. This allows you to debug, spot check your policies, and verify that your policies work as intended.

Log QoS policies
When you encounter problems with the server, you may want to analyze the job logs.

Monitor server transactions
The QoS monitor should be the first point for finding and correcting your QoS problems. It records and allows you to view QoS performance information.

Trace TCP applications
Use a trace command to log several levels of server actions. This can be helpful when you try to determine QoS policy problems.

Order QoS policies
The order of your policies within the file are very important to the success of your quality of service implementation.

Journal QoS policies

QoS includes a journaling function. Journaling allows you to track QoS policy actions, such as when a policy was added, removed or modified. It creates a log of policy actions as long as you have journaling set to ON. This helps you to debug and spot check where policies are not operating as expected. For example, you set a policy to run from 9:00 AM-4:00 PM. You can check the journal log to see if the policy was actually added at 9:00 AM and removed at 4:00 PM.

If journaling is turned on, journal entries are generated anytime a policy is added, removed, or modified. Using these journals, you create a general file on the iSeries server. You can then use the information recorded in your system's journals to determine how your system is being used. This can help you decide to change various aspects of your policies.

Be selective in what you choose to journal. Journaling can be a heavy burden on your system's resources. To start or stop journaling, you use iSeries Navigator. To view the journal logs, you must use the character-based interface.

To start or stop journaling, do the following:

1. In iSeries Navigator, expand your server —> **Network—> IP Policies**.
2. Right-click **Quality of Service** and select **Configuration**.
3. Right-click **QoS** and select **Properties**.

4. Select the **Run Journaling** box to turn journaling on.
5. Deselect the **Run Journaling** box to turn journaling off.

Attention: If the server is already started before you complete the steps above, you must stop and restart the server. Once journaling has been turned on there are two ways to activate it. You can stop and restart the server or perform a server update. Either one of these will reread the policy.conf file and look for the journaling attribute.

Viewing the journal entries on the monitor

To view these journal entries on screen, do the following:

1. At a command prompt on the iSeries server enter: `DSPJRN JRN(QUSRSYS/QQOS)`. Select **Option 5** on the journal entry that you want to view.

Viewing the journal entries through the output file

If you would like to see the journal entries formatted into one folder, view the `MODEL.OUT` file in the `QUSRSYS` directory. By copying the journal entries to the output file, you can easily view the entries by using query utilities such as Query/400 or SQL. You can also write your own HLL programs to process the entries in the output files.

To copy the QoS journal entries to the system-supplied output file:

1. Create a copy of the system-supplied output file QSYS/QATOQQOS into a user library. You can do this by using the Create Duplicate Object (CRTDUPOBJ) command. The following is an example of the CRTDUPOBJ command:

   ```
   CRTDUPOBJ OBJ(QADSPJR4) FROMLIB(Qsys) OBJTYPE(*FILE) TOLIB(userlib) NEWOBj(userfile)
   ```

2. Use the Display Journal (`DSPJRN`) command to copy the entries from the `QUSRSYS/QQOS` journal to the output file created in the previous step. If you attempt to copy the `DSPJRN` into an output file that does not exist, the system creates a file for you, but this file does not contain the proper field descriptions.
   a. `DSPJRN JRN(QUSRSYS/QQOS) JRNCDE((M)) ENTTYP(MP) CMTCYCID(*ALL) OUTPUT(*OUTFILE) OUTFILFMT(*TYPE4) OUTFILE(userlib/userfile)`
   b. `DSPF FILE(userlib/userfile)`

Log QoS server jobs

When you encounter problems with your QoS policies, you should always analyze the iSeries server job logs. The job log contains error messages and other information related to QoS.

Only one QoS job, `QTOQSRVR`, runs in the subsystem `QSYSWRK`. You can view the old and current QoS server job logs from iSeries Navigator.

To view the log, do the following:

1. Expand **Network** and click **IP Policies**.
2. Right-click **Quality of Service**.
3. Select **Diagnostic tools —>QoS Server Log**.

This opens a window which allows you to work with the job.

The following list shows the most important job names, along with a brief explanation of what the job is used for:

QTCP
This job is the base job that starts all the TCP/IP interfaces. If you have fundamental problems with TCP/IP in general, analyze the QTCPIP job log.

QTOQSRVR
This job is the base QoS job that gives you log information specific to QoS. Run a (work spool file) WRKSPLF QTCP and look for the QTOQSRVR log.

To check the work spool file for an error, perform the following tasks:

1. From a command line interface, enter **WRKSPLF QTCP** and press Enter.
2. The Work with All Spool Files window appears. In the User Data column, look for QTOQSRVR to find errors specifically pertaining to the QoS server.
3. Select **Option 5** on the line you want to display. Read through this information and record the Message ID that explains the problem. For example, TCP920C.
4. Press **F3** twice to return to the main menu.
5. From the command line interface, enter **WRKMSGF** and press **Enter**.
6. On the Work with Message File screen, enter the following information and press **Enter**.
   Message File: QTCPMSG
   Library: *LIBL
7. On the Work with Message File screen, select **option 5** to display the message file you want to view and press **Enter**.
8. On the Display Message Descriptions screen, enter the following information:
   Position to: Enter your message ID from number 3 above and press Enter. For example TCP920C.
9. Select **Option 5** on the desired message ID and press **Enter**.
10. On the Select message details to display, select 30 (All of the Above) and press **Enter**.
11. A detailed description of the message appears.

Monitor server transactions

The QoS monitor can help you in the planning phase and the troubleshooting phase of QoS.

You can use the monitor to analyze your IP traffic through the server. This helps you determine where congestion is occurring within your network. The QoS monitor can help you continue to monitor your network so you can adjust your policies as needed.

Planning and maintaining performance
One of the most difficult parts of implementing QoS is determining what performance limits to set in your policies. There is no specific recommendation because every network is different. To help you determine what values are right for you, you may want to use the monitor before you even start any business-specific policies.

Try to create a differentiated services policy without selecting metering to identify how your current network traffic is behaving. Enable this policy and start the monitor. The monitor's results can help you tune your policies to your specific needs. See a sample monitor policy that will identify how your current traffic is behaving.

Troubleshooting performance problems
You can also use the monitor to troubleshoot problems. Using the monitor output, you can determine if the parameters you assigned to a policy are being followed. For some examples of monitor output, visit the QoS scenarios or view all the monitor fields in monitoring.

Monitor current network statistics

Problem

Within the wizards you are asked to set performance limits. These are values that cannot be recommended, since they are based on individual network requirements. To set these limits, you really need to understand your current network performance. Since you are trying to configure quality of service policies, you probably already have a good idea of your current network needs. To determine exact rate limits, such as token bucket rate, you may want to monitor all the traffic on your server so you can better determine what rate limits to set.

Solution

Create a very broad differentiated service policy that does not contain restrictions (no maximum values), and is applied to all interfaces, and all IP addresses. Use the QoS monitor to record data on this policy.

Step 1: Open QoS within iSeries Navigator.

1. In iSeries Navigator, expand your server —> **Network—> IP Policies**
2. Right-click **Quality of Service** and select **Configuration**.
3. Expand **Outbound bandwidth policies**.
4. Right-click **DiffServ** and select **New Policy**. The New DiffServ policy wizard appears.

Step 2: Create a differentiated service policy
Since you want to collect most traffic entering your network you might call the policy **Network**. Use all IP addresses, all ports, all local IP addresses, and all times (if appropriate). Use the following settings throughout the wizard:

```
Name = Network (can be any name you assign)
Client = All IP addresses
Application = All ports
Protocol = All protocols
Schedule = All times
```

iSeries Navigator lists all the differentiated service policies created on your server.

Step 3: Complete a new class of service
While completing the wizard, you are asked to assign a per-hop behavior, performance limits, and out-of-profiling traffic handling. This is defined in a class of service. Choose extremely large values to allow as much traffic flow as possible.

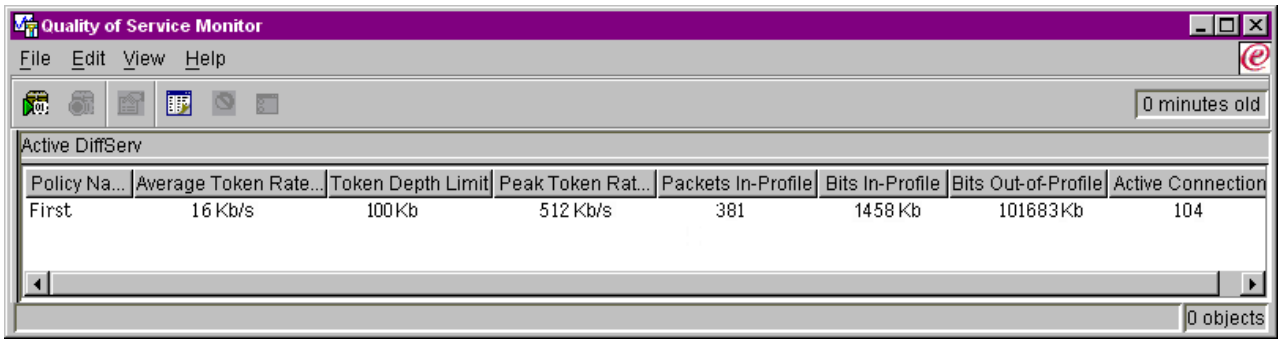
Classes of service actually determine the performance levels that this traffic receives from a router. You could name your class of service **Unlimited**, to show that this traffic receives a higher service. iSeries Navigator lists all the classes of service defined on your server.

Step 4: Monitor your policy
To verify that the traffic is behaving as you configured in the policy, use the monitor.

1. Select the specific Policies folder (DiffServ, IntServ, Server request—>URI, or connection rate).
2. Right-click the policy that you want to monitor and select **Monitor**.

Below is a list of possible monitor output for the policy set above.

Figure 14. Quality of Service Monitor.



Look for the fields that obtain their data from your traffic. Make sure to check the total bits, bits in-profile, packets in-profile, and bits out-of-profile fields. Bits out-of-profile would indicate when traffic exceeds the configured policy values. In a differentiated service policy, the out-of-profile number indicates the number of bytes being dropped. The in-profile packets indicate the number of bytes controlled by this policy (from the time the packet was started to the present monitor output).

What values you assign the average token rate limit field is also important. When packets exceed this limit, the server will begin to drop them. As a result, the bits out-of-profile will increase. This shows you that the policy is behaving as you configured it to behave. To change the amount of bits out-of-profile, you will need to adjust your performance limits. See the monitor section for a description of all the monitor fields.

Step 5: Modify values when needed
After you monitor, you can modify any of the values you previously selected. Right-click on the class of service name you created in this policy. When you select **Properties**, a CoS Properties dialog appears with the values that control your traffic.

Step 6: Monitor the policy again
After seeing the results, use the “guess and check” method to find the best limits for your network needs.

«

Trace TCP applications

Use the QoS trace to work with trace functions and to view the current trace buffer. To run the trace on the server, type `TRCTCPAPP`. Here is a sample of the trace selections to complete:

```
TCP/IP application...........> *QOS
Trace option setting.........> *ON
Maximum storage for trace....> *APP
Trace full action............> *WRAP
Argument lists...............> 'lvl=4'
QoS trace type...............> *ALL
```

The following table introduces the possible parameters to use in a trace. If a setting does not appear on the character-based interface, you must enter them in a command. For example, `TRCTCPAPP APP(*QOS) MAXSTG(1000) TRCFULL(*STOPTRC) ARGLIST('l=4 c=i')`.

Settings	Options
TCP/IP application	QOS
Trace option setting	*ON, *OFF, *END, *CHK
Maximum storage for trace (See 52) (MAXSTG)	1-16000, *APP

Trace full action (See 52) (TRCFULL)	*WRAP, *STOPTRC
Argument list (See 52) (ARGLIST)	Levels: 'lvl=1', 'lvl=2', 'lvl=3', 'lvl=4' Content: 'c=a', 'c=i', 'c=d' , 'c=m', 'c=r', 'c=s'
QoS trace type	*ALL

If you need help interpreting the trace output, see Read the trace output. The trace output page contains sample output with comments to help you interpret its meaning.

Maximum storage for trace

1-16000
This is the maximum storage size for the trace data. The trace either stops or wraps when this size is reached. The default size is 4MB. To specify the default size, select *APP.

***APP**
This is the default option. It tells the application to use its default trace size. The default trace size for the QoS server is 4MB.

Trace full action

***WRAP**
Wraps the trace information when the trace reaches the maximum disk space (trace buffer size). Wrapping will allow the system to overwrite the oldest information in the file, so you continue recording the trace information. If you do not select wrap, then the trace operation stops when the disk is full.

***STOPTRC**
Stops collecting information when the system reaches maximum disk space.

Argument lists

Specifies which error levels and content will be logged. There are two arguments allowed in the TRCTCPAPP command: trace level and trace content. When you specify the trace level and trace content, make sure all attributes are contained in a single set of quotations. For example, `TRCTCPAPP 'l=1 c=a'`

Note: Log levels are inclusive. This means that when you select a log level, all previous log levels are also selected. For example, if you select level 3, then levels 1 and 2 are automatically included.
Trace Levels

Level 1: System errors (SYSERR)
Logs errors that occur in systems operations. If this error occurs, the QoS server cannot continue. For example, a system error may occur if you are running out of system memory or if your system cannot communicate with TCP/IP.

Level 2: Errors between objects (OBJERR)
Logs errors that occur within the QoS server code. For example, an object error may occur because a server operation encounters some unexpected result. This is generally a serious condition that should be reported to service.

Level 3: Specific Events (EVENT)
Logs any QoS operation that has occurred. For example, an event log would record commands and requests. The results are similar to the QoS journaling function.

Level 4: Trace messages (TRACE)
Traces all data being transferred to and from the QoS server. For example, you could use this high-level trace for logging anything that you think would be helpful for debugging problems. This information is helpful to determine where a problem occurred and how to reproduce the problem.

Trace Content

Note: Only specify one content type. If you do not specify what content to trace, then (by default) all content will be traced.

Content = All ('c=a')
Traces all functions of the QoS server. This is the default value. Use this to initially look for a problem.

Content = Intserv ('c=i')
Traces the IntServ operations only. Use this if you determine the problem to be IntServ related.

Content = Diffserv ('c=d')
Traces the DiffServ operations only. Use this if you determine the problem to be DiffServ related.

Content = Monitor ('c=m')
Traces the monitor operations only.

Content = Rate ('c=r')
Traces the inbound connection rate events.

Content = Server ('c=s')
Traces everything but the monitor operations. This can be useful since the monitor trace generates a lot of information that can clutter the trace output unnecessarily.

For more complete information on the TRCTCPAPP command, see TRCTCPAPP (Trace TCP/IP Application) Command Description within the CL commands topic.

Read the trace output

This is not an all-inclusive discussion of how to read your trace output. However, it does highlight the key events to look for in the trace information.

In an **integrated services policy**, the most important event to look for is whether or not the RSVP connection was rejected because a policy for that connection was not found. Here is an example of a successful message:

```
12/15 11:36:14 [336,80] TRCE :.......rpapi_getPolicyData: Found action name vreStnl_kraMoNlCvreStnl for
flow[sess=x.x.x.x:y:z:s, source=x.x.x.x:y]
```

Here is an example of an unsuccessful integrated services connection message:

```
12/15 11:36:14 [336,80] TRCE :.......rpapi_getPolicyData: Unable to find action name for flow
[sess=x.x.x.x:y]
```

For a **differentiated services policy**, the most important messages show if the server loaded a policy rule or if an error occurred in the policy configuration file.

```
Example:
01/11 14:07:52 [376,57] TRCE :......KernelAddPolicyRule: Installing rule = timed_42ring.
01/11 14:07:52 [376,57] EVNT :.......create_tcp_resv: No value in config file for
DiffServInProfilePeakRate, defaulted to 100000 00.
01/11 14:07:52 [376,57] TRCE :.......create_tcp_resv: Create resv - bRate: 537395 5722SS1 V5R1M0
010525 TRCTCPAPP Output RS004 Date-01/11/01 Time-14:08:03 Page-6
01/11 14:07:52 [376,57] TRCE :.......create_tcp_resv: bDepth: 32768
01/11 14:07:52 [376,57] TRCE :.......create_tcp_resv: peakR: 10000000
01/11 14:07:52 [376,57] TRCE :.......create_tcp_resv: m: 128
01/11 14:07:52 [376,57] TRCE :.......create_tcp_resv: M: 41452
01/11 14:07:52 [376,57] TRCE :.......create_tcp_resv: mark(TOS): a0
01/11 14:07:52 [376,57] TRCE :.......create_tcp_resv: flags: 15
01/11 14:07:52 [376,57] TRCE :.......create_tcp_resv: flowspe.form = 1, QOS_FORMAT_DS = 1
```

You may also have messages showing that the tags in the policy configuration file were incorrect. Here are some sample messages:

```
12/15 11:36:14 [336,80] TRCE :.......rpapi_getPolicyData: Unknown attribute %s in ServicePolicy-Ignoring.
12/15 11:36:14 [336,80] TRCE :.......rpapi_getPolicyData: Unknown attribute %s in Priority
Mapping-Ignoring.
```

Note: The % sign is a variable that represents an unrecognized tag.

Related information for QoS

There are many other sources of information on quality of service in the industry. Review the latest RFCs, white papers, Redbooks(TM), and other sources to receive general information about QoS. Here are some sources to consider:

Non-IBM sources

RFC 1349

This RFC discusses the new definition of the TOS field in an IP packet header.

RFC 2205

This RFC covers the definition of Resource ReSerVation Protocol (RSVP).

RFC 2210

This RFC covers the use of RSVP with IETF Integrated Services.

RFC 2474

This RFC covers the definition of the Differentiated Services Field (DS Field).

RFC 2475

This RFC covers the architecture of differentiated services.

IBM(R) Redbooks

TCP/IP More Cool Things than Ever

This manual provides sample scenarios that demonstrate common solutions with example configurations. The information in this manual helps you plan, install, tailor, configure, and troubleshoot TCP/IP on your iSeries server. It does not specifically cover Quality of service yet, but it does go through LDAP directory server information.

TCP/IP Tutorial and Technical Overview

This manual provides an introduction as well as a reference to the Transmission Control Protocol/Internet Protocol (TCP/IP) suite of protocols and applications. You will find Quality of service within *Part 3. Advanced concepts and new technologies* under Chapter 22.

Related iSeries Information Center topics

Directory services (LDAP)
View this topic to obtain directory server basics, configuration, administration, and troubleshooting. The directory services topic will also give you additional resources for configuring your directory server.

Printed in U.S.A.