IBM Systems - iSeries

Networking
Quality of service

Version 5 Release 4

IBM Systems - iSeries

Networking
Quality of service

Version 5 Release 4

Note

Before using this information and the product it supports, read the information in "Notices," on page 69.

Fifth Edition (Februray 2006)

This edition applies to version 5, release 4, modification 0 of IBM i5/OS (product number 5722-SS1) and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CISC models.

Contents

Quality of service

The iSeries™ quality of service (QoS) solution enables policies to request network priority and bandwidth for TCP/IP applications throughout the network.

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.

Packet priority is important to you if you send applications that need predictable and reliable results, such as multimedia. QoS policies on the iSeries server can also limit data leaving your server, manage connection requests, and control server load. The QoS server must be running to activate the intrusion detection policy.

It is important to understand QoS before you start configuring policies.

What's new for V5R4

New intrusion detection function

For V5R4, the quality of service (QoS) server provides intrusion detection capabilities through an intrusion detection policy. To activate this new policy, the QoS server must be running. Using the QoS intrusion detection policy, you can detect intrusions, create auditing records, and send messages to indicate a possible intrusion attempt. For more information, see Intrusion detection function.

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.

Printable PDF

Use this to view and print a PDF of this information.

To view or download the PDF version of this document, select Quality of service (about 525 KB).

Saving PDF files

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

1. Right-click the PDF in your browser (right-click the link above).
2. Click the option that saves the PDF locally.
3. Navigate to the directory in which you want to save the PDF.
4. Click **Save**.

Downloading Adobe Reader

You need Adobe Reader installed on your system to view or print these PDFs. You can download a free copy from the Adobe Web site (www.adobe.com/products/acrobat/readstep.html) .

Concepts

If you are new to quality of service (QoS), you can read some basic QoS concepts. This will give you an overview of how QoS works and how QoS functions work together.

Before attempting to do QoS, it is suggested that you research the topic and make sure that this service will meet your needs. QoS terms can be found in multiple sources, so this topic will only discuss the basic terminology.

To carry out QoS, you configure policies using wizards in iSeries Navigator. A *policy* is a set of rules that designate an action. The policy basically states which client, application, and schedule (which you designate) must receive a particular service. You can ultimately configure three policy types:

- Differentiated service
- Integrated service
- Inbound admission

Differentiated service and *integrated service* are considered outbound bandwidth policies. Outbound policies limit data leaving your network and help control server load. The rates you set within an outbound policy control how and what data is or is not limited within the server. Both outbound policy types might require a service-level agreement (SLA) with your Internet service provider (ISP).

Inbound admission policies control connection requests coming into your network from some outside source. Inbound policies are not dependent on a service-level from your ISP. To decide which policy you need to use, evaluate the reasons why you want to use QoS and consider the role of your iSeries server.

One of the most important parts of carrying out QoS is your server itself. Not only do you need to understand the QoS concepts, but you also need to be aware of the role your server plays in these concepts. The iSeries server can only act as a client or a server, not a router. For example, an iSeries server acting as a client, can use differentiated service policies to ensure that information requests to other servers are given higher priority through the network. An iSeries server acting as a server, can use an inbound admission policy to limit Uniform Resource Identifier (URI) requests accepted by the server.

Related concepts

"Service-level agreement" on page 49
This topic points out some of the important aspects of a service-level agreement (SLA) that might affect your quality of service (QoS) implementation.

Related reference

"Related information for QoS" on page 67
Listed here are the IBM® Redbooks™ (in PDF format), Web sites, and information center topics that relate to the quality of service (QoS) topic. You can view or print any of the PDFs.

Differentiated service

This is the first type of outbound bandwidth policy you can create on your server. Differentiated service divides your traffic into classes. To carry out a differentiated service policy, you need to determine how you want to classify your network traffic and how to handle the different classes.

Related concepts

"QoS sendmsg() API extensions" on page 23
The sendmsg() function is used to send data, ancillary data, or a combination of these through a connected or unconnected socket.

"Token bucket and bandwidth limits" on page 10
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.

"Class of service" on page 13
When you create a differentiated service policy or an inbound admission policy, you also create and use a class of service.

"Scenario: Limit browser traffic" on page 28
You can use quality of service (QoS) to control traffic performance. Use a differentiated service policy to either limit or extend an application's performance within your network.

"Scenario: Secure and predictable results (VPN and QoS)" on page 32
If you are using a virtual private network (VPN), you can still create quality of service (QoS) policies. This example shows the two being used together.

Related reference

"Use codepoints to assign per-hop behaviors" on page 14
Quality of service (QoS) uses the recommended codepoints to assign per-hop behaviors to traffic.

"Configure QoS with wizards" on page 51
To configure quality of service (QoS) policies, you must use the QoS wizards located in iSeries Navigator.

Related information

Manage addresses and ports for your HTTP server (powered by Apache)

Prioritized classes: How to classify network traffic

Differentiated service identifies traffic as *classes*. The most common classes are defined using client IP addresses, application ports, server types, protocols, local IP addresses, and schedules. All traffic that conforms to the same class is treated equally.

For more advanced classification, you can specify server data to set different levels of service for some of your iSeries applications. Using server data is optional, but might be helpful when you want to classify at a lower level. Server data is based on two types of application data: *application token* or *Uniform Resource Identifier (URI)*. If traffic matches the token or URI you specify in the policy, the policy will be applied to the outbound response, thus giving the outbound traffic, whatever priority is specified in the differentiated service policy.

Use application token with differentiated service policies

Using application data tells the policy to respond to specific parameters (token and priority) passed by the application to the server through the sendmsg() application programming interface (API). This setting is optional. If you do not need this level of granularity in your outbound policies, select **All tokens** in the wizard. You can match an application's token and priority with a specific token and priority set in the outbound policy, if you want to. In the policy, there are two parts to setting the application data, which include the token and the priority.

- What is an application token?

  An application token is any character string that can represent a defined resource, such as `myFTP`. The token you specify in the quality of service (QoS) policy is matched against the token provided by the outbound application. The application provides the token value through the sendmsg() API. If the tokens match, the application traffic is included in the differentiated service policy.

  To use an application token in a differentiated service policy, follow these steps:

  1. From the QoS configuration window, right-click **DiffServ** and select **New Policy**. Start the wizard.
  2. On the Server Data Request page, select **Selected application token**.
  3. To create a new token, click **New**. The New URI window appears.
  4. In the **Name** field, enter a meaningful name for the application token.

5. In the **URI** field, delete the (/) and enter the application token (a string of not more than 128 characters). For example, `myFTPapp`, rather than the typical URI.

- What is an application priority?

  The application priority you specify is matched against the application priority provided by the outbound application. The application provides the priority value using the sendmsg() API. If the priorities match, the application traffic is included in the differentiated service policy. All traffic defined in the differentiated service policy will still receive the priority given to the entire policy.

When you specify application token as the application data type, the application providing this information to the server must be specifically coded to use the sendmsg() API. This is done by the application programmer. The application's documentation should provide valid values (token and priority), which the QoS administrator will use in the differentiated service policy. The differentiated service policy then applies its own priority and classification to traffic that matches the token set in the policy. If the application does not have values that match the values set in the policy, you must either update the application or use different application data parameters for the differentiated service policy.

Use a URI with differentiated service policies

When creating a differentiated service policy, the wizard allows you to set server data information, as discussed in "Use application token with differentiated service policies" section. Although the fields in the wizard prompt you for an application token, you can instead specify a relative URI. Again, this is optional. If you do not need this level of granularity in your outbound policies, select **All tokens** in the wizard. You can match a specific URI set in the outbound policy if you want to.

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 segments are valid relative URI examples:

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

Before you set up a differentiated service policy that uses URLs, you must ensure that the application port assigned for the URI matches the Listen directive enabled for the Fast Response Cache Accelerator (FRCA) in the Apache Web Server configuration. To change or view the port for your HTTP server, see Manage addresses and ports for your HTTP server (powered by Apache).

FRCA identifies the URI for each outbound HTTP response. It compares the URI related to the outbound response to the URI defined in each differentiated service policy. The first policy with a token string (URI) that best matches the URI identified by FRCA, is applied to all responses for the URI.

Related concepts

"QoS sendmsg() API extensions" on page 23
The sendmsg() function is used to send data, ancillary data, or a combination of these through a connected or unconnected socket.

Setting priorities: How to handle the classes

After traffic is classified, differentiated service also requires a per-hop behavior to define how to handle the traffic.

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 information in the IP header's type of service octet field. The type of service octet field was redefined in the Request for Comments (RFC) 1349 and OS/400® V5R1 operating system. A per-hop behavior is the forwarding behavior that a packet receives at a network node. It is represented by a value known as a *codepoint*. Packets can be marked either at the server or at other parts of the network, such as a router. For a packet to retain the service requested, every network node must be differentiated service (DiffServ)-aware. That is, the equipment must be able to enforce per-hop behaviors. To enforce per-hop behavior treatment, the network node must be able to use queue scheduling and outbound priority management. See the “Traffic conditioners” topic for more information about what it means to be DiffServ-aware.

If your packet passes through a router or switch that is not DiffServ-aware, it will lose its level of service at that router. The packet is still handled, but it might experience unexpected delay. On your iSeries server, you can use the pre-defined per-hop behavior codepoints or you can define your own codepoint. It is not recommended that you create your own codepoints for use outside your private network. If you do not know which codepoints to assign, review “Use codepoints to assign per-hop behaviors” on page 14 topic.

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

Differentiated service can also be used to throttle traffic leaving a server. This means that your iSeries server really uses differentiated service to limit performance. Limiting a less-critical application allows a mission-critical application to exit your private network first. When you create a class of service for this 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 quality of service (QoS) function of iSeries Navigator gives you more specific information about these limits.

Traffic conditioners

To use quality of service (QoS) policies, network equipment (like routers and switches) must have the capability for traffic conditioners. Traffic conditioners refer to classifiers, meters, markers, shapers, and droppers.

If the network equipment has all the traffic conditioners, then it is considered *DiffServ-aware*.

Note: These hardware requirements are not iSeries specific. You will not see these terms used in the QoS interface, because the server cannot control external hardware. Outside a private network, hardware needs to have the ability to handle general QoS requirements. Check with the specific equipment manuals to make sure that they can handle differentiated service requirements. It is also recommended that you research general QoS concepts and prerequisites before implementing policies.

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

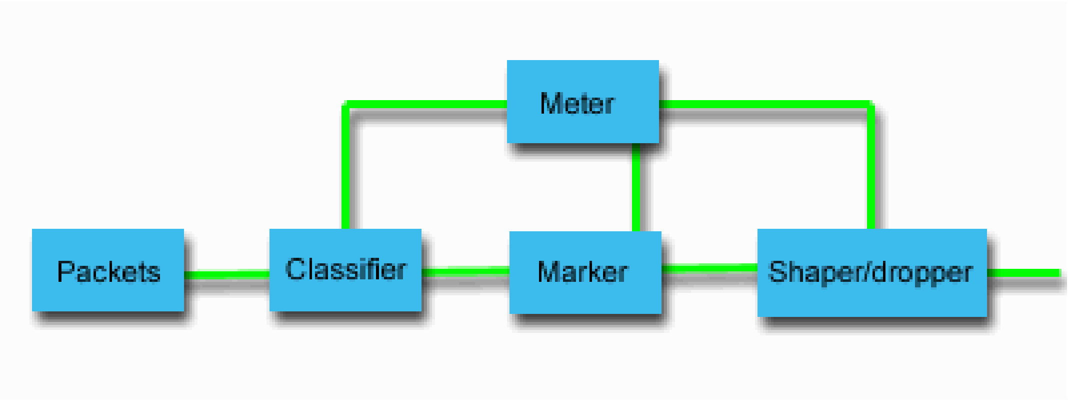


Figure 1. Traffic conditioners

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 behavior aggregate (BA) classifies packets, based exclusively on the differentiated services codepoint. The multi-field (MF) 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, Uniform Resource Identifier (URI), server type and destination port numbers.

Meters

Traffic meters measure whether 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. 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 routers might discard packets 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.

Related concepts

"Network hardware and software" on page 50
The capabilities of your internal equipment and other equipment outside your network have enormous effects on quality of service (QoS) results.

Integrated service

The second type of outbound bandwidth policy you can create is an integrated service policy. Integrated service provides the capability for IP applications to request and reserve bandwidth using the ReSerVation Protocol (RSVP) and quality of service (QoS) APIs.

Integrated service policies use the RSVP protocol and the Resource Reservation Setup Protocol (RAPI) API (or qtoq socket API) to guarantee an end-to-end connection. This is the highest level of service you can designate; however, it is also the most complex.

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

Integrated service 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 RSVP protocol and RAPI API, or the RSVP protocol and qtoq QoS sockets APIs.

Every node that your traffic travels through must have the ability to use the RSVP protocol. The routers provide QoS 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 must be RSVP-enabled. For example, your traffic is routed based on resources, not on which paths have RSVP-enabled routers. Crossing routers that are not RSVP-enabled might 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.

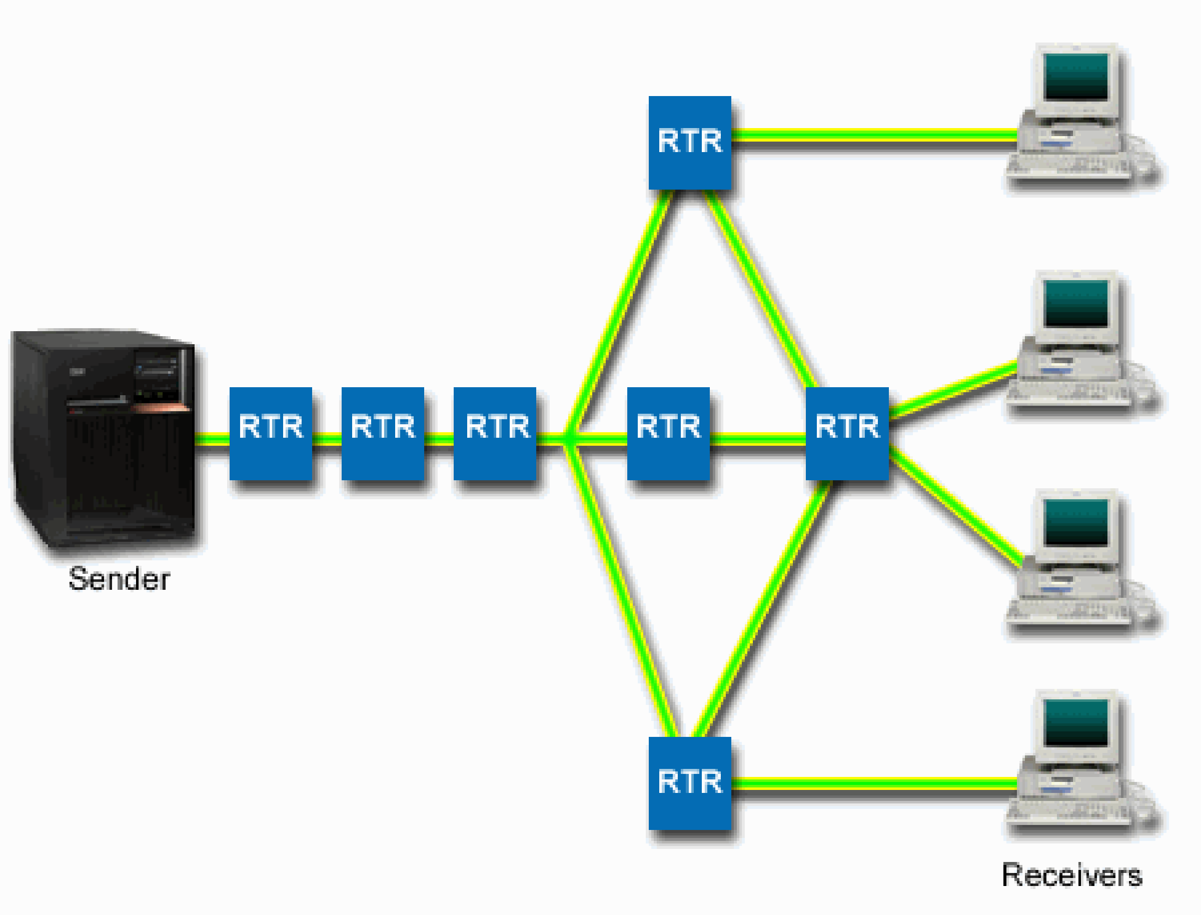


Figure 2. RSVP path between client and server

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 service 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 QoS. Applications used in an integrated service 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.

As packets arrive and attempt to leave your network, your server determines whether 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 allows. These values are referred to as performance limits. If the packets remain within the server's limits, the packets conform and are sent out. In integrated services, each connection is granted its own token bucket.

Integrated service using differentiated service markings

If you are unsure that the entire network can guaranteed an RSVP connection, you can still create an integrated service policy. However, if the network resources cannot use RSVP, the connection cannot be guaranteed. In this situation, you might want to apply a codepoint to the policy. This codepoint is typically used in differentiated service policies to give a class of service to traffic. Even if the connection is not guaranteed, this codepoint will attempt to give the connection some priority.

Related concepts

"QoS APIs" on page 16
You can read this topic to learn about protocols, APIs, and requirements for a router that is enabled for the ReSerVation Protocol (RSVP). The current quality of service (QoS) APIs include the RAPI API, the qtoq socket API, the sendmsg() API, and the monitor APIs.

"Integrated service using differentiated service markings" on page 10
Use differentiated service markings in an integrated service policy to maintain priority of packets sent in a mixed environment.

"Scenario: Predictable B2B traffic" on page 38
If you need predictable delivery and still need to request a reservation, you also use an integrated service policy. This example uses a controlled load service.

"Scenario: Dedicated delivery (IP telephony)" on page 42
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, guaranteed service is used.

Traffic control functions

Traffic control functions only apply to integrated service and are not specific to iSeries.

You will not see these terms used in the quality of service (QoS) interface, because the server cannot control external hardware. Outside a private network, hardware needs to have the ability to handle general QoS requirements. The general router requirements for IntServ policies are discussed in the following section. It is suggested that you research general QoS concepts and prerequisites before implementing policies.

To get predictable results, you need to have hardware that is enabled by ReSerVation Protocol (RSVP), along the traffic's path. Routers must have certain traffic control functions in order to use RSVP. 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 cannot be used as a router at this time. Check with your network equipment manuals, to verify that they can handle QoS requirements.

Traffic control functions can include the following functions:

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 on 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.

Related concepts

"QoS APIs" on page 16
You can read this topic to learn about protocols, APIs, and requirements for a router that is enabled for the ReSerVation Protocol (RSVP). The current quality of service (QoS) APIs include the RAPI API, the qtoq socket API, the sendmsg() API, and the monitor APIs.

Related reference

"Related information for QoS" on page 67
Listed here are the IBM Redbooks (in PDF format), Web sites, and information center topics that relate to the quality of service (QoS) topic. You can view or print any of the PDFs.

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 of 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 quality of service (QoS) enabled with support for Integrated services. You will need to check the router's specifications to see if they provide QoS 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 ReSerVation Protocol (RSVP) capabilities to assure delivery. When you assign the token bucket limits and bandwidth limits, you are defining your guaranteed service. Guaranteed service can only be applied to applications using TCP.

Related concepts

"Scenario: Predictable B2B traffic" on page 38
If you need predictable delivery and still need to request a reservation, you also use an integrated service policy. This example uses a controlled load service.

"Scenario: Dedicated delivery (IP telephony)" on page 42
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, guaranteed service is used.

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 amount of information your server can process at any given time. If an application is sending your server information faster than the server can send the data out of the network, the buffer fills up. Any data packets exceeding this limit are treated as out-of-profile. Integrated service policies are the exception to this rule. You can select `do not limit`, which will allow a ReSerVation Protocol (RSVP) connection request. For all other policies, you can determine how to handle out-of-profile traffic. The maximum token bucket size is 1 GB.

Token rate limit

The rate limit specifies the long term data rate or the number of bits per second allowed into a network. The quality of service (QoS0 policy looks at the requested bandwidth and compares it with the rate and flow limits for this policy. If the request causes 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. This value can vary between 10 Kbps to 1 Gbps. You can also set this to `do not limit`. When you assign `do not limit` to the rate, you are making the available resources the limit.

Tip: To determine what limits to set, you might 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. The scenario about monitoring current network statistics shows one way to collect the total rates your application and network currently use. Using these results, you can reduce the limits appropriately.

To view real-time monitor data instead of a particular data collection, just open the monitor. The monitor gives real-time statistics on all active policies.

Related concepts

"Differentiated service" on page 2
This is the first type of outbound bandwidth policy you can create on your server. Differentiated service divides your traffic into classes. To carry out a differentiated service policy, you need to determine how you want to classify your network traffic and how to handle the different classes.

"Scenario: Monitor current network statistics" on page 46
Within the wizards, you are asked to set performance limits. These values cannot be recommended because they are based on individual network requirements.

Integrated service using differentiated service markings

Use differentiated service markings in an integrated service policy to maintain priority of packets sent in a mixed environment.

A mixed environment occurs when an integrated service reservation travels through different routers that do not support integrated service reservations but do support differentiated service. Because your traffic passes through different domains, service-level agreements, and equipment capabilities, you might not always get the service you intend.

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

No signaling

In addition to using markings, you can also use the *no-signal* function. When you select this function, the no-signal versions of the APIs 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 signaling 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 RSVP.

The no-signal function is specified within the integrated service policy. To designate no signal, perform the following steps:

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 the required IntServ policy name and select **Properties**. The IntServ Properties window opens.
5. Select the **Traffic Management** tab to disable or enable signaling. This is also where you edit the schedule, client, applications, and traffic management.

Related concepts

"Class of service" on page 13
When you create a differentiated service policy or an inbound admission policy, you also create and use a class of service.

"Integrated service" on page 6
The second type of outbound bandwidth policy you can create is an integrated service policy. Integrated service provides the capability for IP applications to request and reserve bandwidth using the ReSerVation Protocol (RSVP) and quality of service (QoS) APIs.

Inbound admission policy

The inbound admission policy is used to control connection requests coming into your network.

The inbound policy is used to restrict traffic attempting to connect to your server. You can restrict access by client, Uniform Resource Identifier (URI), application, or local interface on your iSeries server. In addition, you can enhance server performance by applying a class of service to inbound traffic. You define this policy through the Inbound admission wizard in iSeries Navigator.

There are three components to an inbound policy which require more information. They include URIs to restrict traffic, connection rates defined in a class of service, and priority queues to order successful connections. For more information, see "URI," "Connection rate" on page 12, and "Weighted priority queues" on page 12.

URI

You might consider using an inbound policy to restrict HTTP traffic connecting to your Web server. In this circumstance, you might create an inbound admission policy that restricts traffic by a specific URI. URI request rate is part of a solution to help protect servers against overload. Designating specific URIs will apply 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.

Specifying a URI allows the inbound policy to examine content, not just packet headers. The content examined is a URI name. For iSeries, you can use the relative URI name (for example, **`/products/clothing`**). The following examples 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 segments are valid relative URI examples:

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

Notes:

1. When using a URI, you must specify the protocol as TCP. In addition, the port and IP address must match the port and IP address configured for your HTTP server. This is typically port 80.
2. There is an implicit wildcard when you specify a URI. For example, /software will include anything within the software directory.
3. Do not use an * in the URI. It is not a valid character.
4. URI information can be used in either inbound policies or differentiated service (outbound) policy.

Before you set up an inbound policy that uses URIs, you must ensure that the application port assigned for the URI matches the Listen directive enabled for Fast Response Cache Accelerator (FRCA) in the Apache Web Server configuration. To change or view the port for your HTTP server, see Manage addresses and ports for your HTTP server (powered by Apache).

Connection rate

As part of the inbound admission policy, you also must select a class of service. This class of service defines connection rates that act as admission control to limit the connections accepted by the server.

Connection rate limits accept or deny a new packet, based on the average number of connections per second and the maximum number of instantaneous connections defined in the policy you create. These connection limits consist of average rate and burst limit, which the wizards in iSeries navigator will prompt you to enter. When incoming connection requests reach the server, the server analyses 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 packet is within the policy limits, it is placed into the queue.

Use the above information as you complete the Inbound admission wizard. In iSeries Navigator, you can also use the associated Help to refer to similar information as you complete the policy.

Weighted priority queues

As part of inbound control, you can specify the priority in which connection requests are handled after they have been evaluated by the policies. By assigning a weight to a priority queue, you are essentially controlling the queue's response time after a connection has arrived. If queued, the connection will be handled in order of queue priority (high, medium, low, or best effort). If you are unsure of what weights to assign, use the default values. The sum of all the weights must equal 100. For example, If 25 is specified for all priorities, then all queues are treated equally. Suppose that you specify the following weights: High (50), Medium (30), Low (15), and Best effort (5). The accepted connections include:

- 50% high priority connections
- 30 % medium priority connections
- 15% low priority connections
- 5% best effort priority connections

Related concepts

“Class of service”
When you create a differentiated service policy or an inbound admission policy, you also create and use a class of service.

“Average connection rate and burst limits” on page 16
Connection rates and burst limits, together, are known as *rate limits*. These rate limits help restrict inbound connections trying to enter your server. Rate limits are set in a class of service used with inbound admission policies.

Class of service

When you create a differentiated service policy or an inbound admission policy, you also create and use a class of service.

Differentiated service policies and Inbound admission policies use a class of service to group traffic into classes. Even though most of this happens through hardware, you control how you group traffic and what priority the traffic must receive.

As you carry out quality of service (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 might be reused by policies. When you define the class of service, you specify if it can be applied to outbound, inbound, or both policy types. If you select both (outbound and inbound), then a differentiated service policy and an inbound admission policy can use that class of service.

The settings within the class of service depend on whether the class of service is used for inbound, outbound, or both types of policies. When you create the class of service, you might encounter the following requirements:

Codepoint marking
QoS uses the recommended codepoints to assign per-hop behaviors to traffic. Routers and switches use these codepoints to give traffic priority levels. Your server cannot use these codepoints, because it does not act as a router. You must determine which codepoints to use, based on your individual network needs. Consider what applications are most important to you and what policies must 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.

Traffic metering
QoS uses rate control 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 limits” on page 10 for more information about these specific values.

Out-of-profile traffic
The final portion of a class of service is out-of-profile handling. When you assign the rate control limits above, you set values to restrict traffic. When traffic exceeds these restrictions, the packets are considered out-of-profile. The information in a class of service tells the server whether to drop UDP traffic and reduce TCP congestion window, shape, or remark out-of-profile packets.

Drop UDP packets or reduce TCP congestion window: If you decide to drop and adjust out-of-profile packets, the UDP packets are dropped. However, the TCP congestion window is reduced so that data rate complies with the token bucket rate. The number of packets that can be sent into the network at any given time decreases, and as a result reduces congestion.

Delay (Shape): If you delay the out-of-profile packets, they are shaped to conform to your defined handling characteristics.

Re-mark with DiffServ codepoint: If you re-mark out-of-profile packets with a codepoint, they are reassigned a new codepoint. The packets are not throttled to meet your handling characteristics, just re-marked. When you assign these handling instructions in the wizard, click Help for more specific information.

Priority

You can prioritize the connections that are made to your server by different inbound admission control policies. This allows you to define the order in which completed connections are handled by your server. You can choose high, medium, low, or best effort.

Related concepts

"Integrated service using differentiated service markings" on page 10
Use differentiated service markings in an integrated service policy to maintain priority of packets sent in a mixed environment.

"Inbound admission policy" on page 11
The inbound admission policy is used to control connection requests coming into your network.

"Differentiated service" on page 2
This is the first type of outbound bandwidth policy you can create on your server. Differentiated service divides your traffic into classes. To carry out a differentiated service policy, you need to determine how you want to classify your network traffic and how to handle the different classes.

Related reference

"Use codepoints to assign per-hop behaviors"
Quality of service (QoS) uses the recommended codepoints to assign per-hop behaviors to traffic.

Use codepoints to assign per-hop behaviors

Quality of service (QoS) uses the recommended codepoints to assign per-hop behaviors to traffic.

In the Class of service wizard, you will need to assign a per-hop behavior to your policy. You must 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 might be assigned higher priority. The most important thing is to be consistent with your markings, so that you get the results you expect. For example, policies that hold similar importance might use similar codepoints so that you get consistent results for those policies. If you are unsure which codepoint to assign, use trial and error. You can create test policies, monitor these policies, and make adjustments accordingly.

The tables in the following sections display the suggested codepoints, which are based on industry standards. Most Internet service providers (ISPs) support the industry-standard codepoints, and you can verify whether your ISP supports these codepoints. Across domains, every ISP must agree to support QoS requests. 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 might waste your resources. QoS policies allow you to negotiate service-levels with your ISP, which might decrease network service costs. You can also create your own codepoints; however, it is not suggested for external use. Your own codepoints might be best used in a testing environment.

Expedited forwarding

Expedited forwarding is one type of 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.

Table 1. Recommended codepoints: Expedited forwarding

Expedited forwarding
101110

Note: There is typically 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 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.

Table 2. Recommended codepoints: Class selector

Class selector
Class 0 - 000000
Class 1 - 001000
Class 2 - 010000
Class 3 - 011000
Class 4 - 100000
Class 5 - 101000
Class 6 - 110000
Class 7 - 111000

Assured forwarding

Assured forwarding is divided into four per-hop behavior classes, each of which has 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 the policy the highest priority. A low drop level means that the packets in this policy have the lowest chance of being dropped in this particular class level.

Table 3. Recommended codepoints: Assured forwarding

Assured forwarding
Assured forwarding, Class 1, Low - 001010
Assured forwarding, Class 1, Medium - 001100
Assured forwarding, Class 1, High- 001110
Assured forwarding, Class 2, Low - 010010
Assured forwarding, Class 2, Medium - 010100
Assured forwarding, Class 2, High - 010110
Assured forwarding, Class 3, Low - 011010
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

Related concepts

"Differentiated service" on page 2
This is the first type of outbound bandwidth policy you can create on your server. Differentiated service divides your traffic into classes. To carry out a differentiated service policy, you need to determine how you want to classify your network traffic and how to handle the different classes.

"Class of service" on page 13
When you create a differentiated service policy or an inbound admission policy, you also create and use a class of service.

Average connection rate and burst limits

Connection rates and burst limits, together, are known as *rate limits*. These rate limits help restrict inbound connections trying to enter your server. Rate limits are set in a class of service used with inbound admission policies.

Connection burst rate

The burst rate size determines the buffer capacity, which holds connection bursts. Connection bursts might enter the server at a faster rate than it can handle or that you might 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 Uniform Resource Identifier (URI) requests allowed into a server. If a request causes the server to exceed the limits you set, the server denies the request. The average connection request limit is measured in connections per second.

Tip: To determine what limits to set, you can run the monitor. The scenario about monitoring current network statistics contains a sample policy that might help you collect most data travelling on your server. Using these results, you can adjust the limits appropriately.

To view real-time monitor data instead of a particular data collection, open the monitor. The monitor gives real-time statistics on all active policies.

Related concepts

"Inbound admission policy" on page 11
The inbound admission policy is used to control connection requests coming into your network.

"Scenario: Monitor current network statistics" on page 46
Within the wizards, you are asked to set performance limits. These values cannot be recommended because they are based on individual network requirements.

QoS APIs

You can read this topic to learn about protocols, APIs, and requirements for a router that is enabled for the ReSerVation Protocol (RSVP). The current quality of service (QoS) APIs include the RAPI API, the qtoq socket API, the sendmsg() API, and the monitor APIs.

Most QoS policies require the use of an API. The following APIs might be used in conjunction with either a differentiated service or integrated service policies. There are also a number of APIs to use with the QoS monitor.

- "Integrated service APIs"
- "Differentiated service APIs" on page 17
- "Monitor APIs" on page 18

Integrated service APIs

The RSVP, along with the RAPI APIs or qtoq QoS sockets APIs, performs your integrated service reservation. Every node that your traffic travels through must have the ability to use RSVP. The ability to carry out integrated services policies is often referred to as *RSVP-enabled*. The traffic control functions can be used to determine which router functions are needed to use RSVP.

RSVP 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. Because 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 socket 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 message handling. You need a thorough understanding of the RSVP protocol and the contents of Internet RFC 2205.

qtoq socket APIs

You can use the qtoq QoS socket APIs to simplify the work required to use the RSVP protocol on the iSeries system. The qtoq socket APIs call the RAPI APIs and perform some of the more complex tasks. The qtoq socket APIs are not as flexible as the RAPI APIs, but they provide the same functions with less effort. The no-signal versions of the APIs allow you to write the following applications:

- 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 signaling 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.

Differentiated service APIs

Note: The sendmsg() API is used for certain differentiated service policies that define a specific application token. When you create a differentiated service spolicy, you can (optionally) provide application characteristics (token and priority). This is an advanced policy definition, and if not used, this API can be ignored. However, remember that routers and other servers along the network still need to be aware of differentiated service.

If you decide to use an application token in a differentiated service policy, the application providing this information must be specifically coded to use the sendmsg() API. This is done by the application programmer. The application's documentation must provide valid values (token and priority), which the QoS administrator will use in the differentiated service policy. The differentiated service policy then applies its own priority and classification to traffic which matches the token set in the policy. If the application does not have values which match the values set in the policy, either the application must be changed, or you need to use different application data parameters for the differentiated service policy.

The following information briefly describes the server data parameters: application token and application priority.

What is an application token?

An *application token* is a Uniform Resource Identifier (URI) that represents a defined resource. The token you specify in the QoS policy is matched against the token provided by the outbound application. The application provides the token value by using the sendmsg() API. If the tokens match, the application traffic is included in the differentiated service policy.

What is an application priority?

The application priority you specify is matched against the application priority provided by the outbound application. The application provides the priority value by using the sendmsg() API. If the priorities match, the application traffic is included in the differentiated service policy. All traffic defined in the differentiated service policy will still receive the priority given to the entire policy.

For more information about the DiffServ policy type, see “Differentiated service” on page 2.

Monitor APIs

The Resource Reservation Setup Protocol APIs include the monitor APIs. The APIs that apply to the monitor will have the word *monitor* in the title. For example, *QgyOpenListQoSMonitorData*. The following list briefly describes each monitor API:

- QgyOpenListQoSMonitorData (Open List of QoS Monitor Data) gathers information related to QoS services.
- QtoqDeleteQoSMonitorData (Delete QoS Monitor Data) deletes one or more sets of collected QoS monitor data.
- QtoqEndQoSMonitor (End QoS Monitor) stops gathering information related to QoS services.
- QtoqListSavedQoSMonitorData (List Saved QoS Monitor Data) returns a list of all collected monitor data that was saved previously.
- QtoqSaveQoSMonitorData (Save QoS Monitor Data) saves a copy of the collected QoS monitor data for future use.
- QtoqStartQoSMonitor (Start QoS Monitor) gathers information related to QoS services.

Related concepts

“Integrated service” on page 6
The second type of outbound bandwidth policy you can create is an integrated service policy. Integrated service provides the capability for IP applications to request and reserve bandwidth using the ReSerVation Protocol (RSVP) and quality of service (QoS) APIs.

“Traffic control functions” on page 8
Traffic control functions only apply to integrated service and are not specific to iSeries.

“Scenario: Predictable B2B traffic” on page 38
If you need predictable delivery and still need to request a reservation, you also use an integrated service policy. This example uses a controlled load service.

“Network hardware and software” on page 50
The capabilities of your internal equipment and other equipment outside your network have enormous effects on quality of service (QoS) results.

Related reference

RAPI API

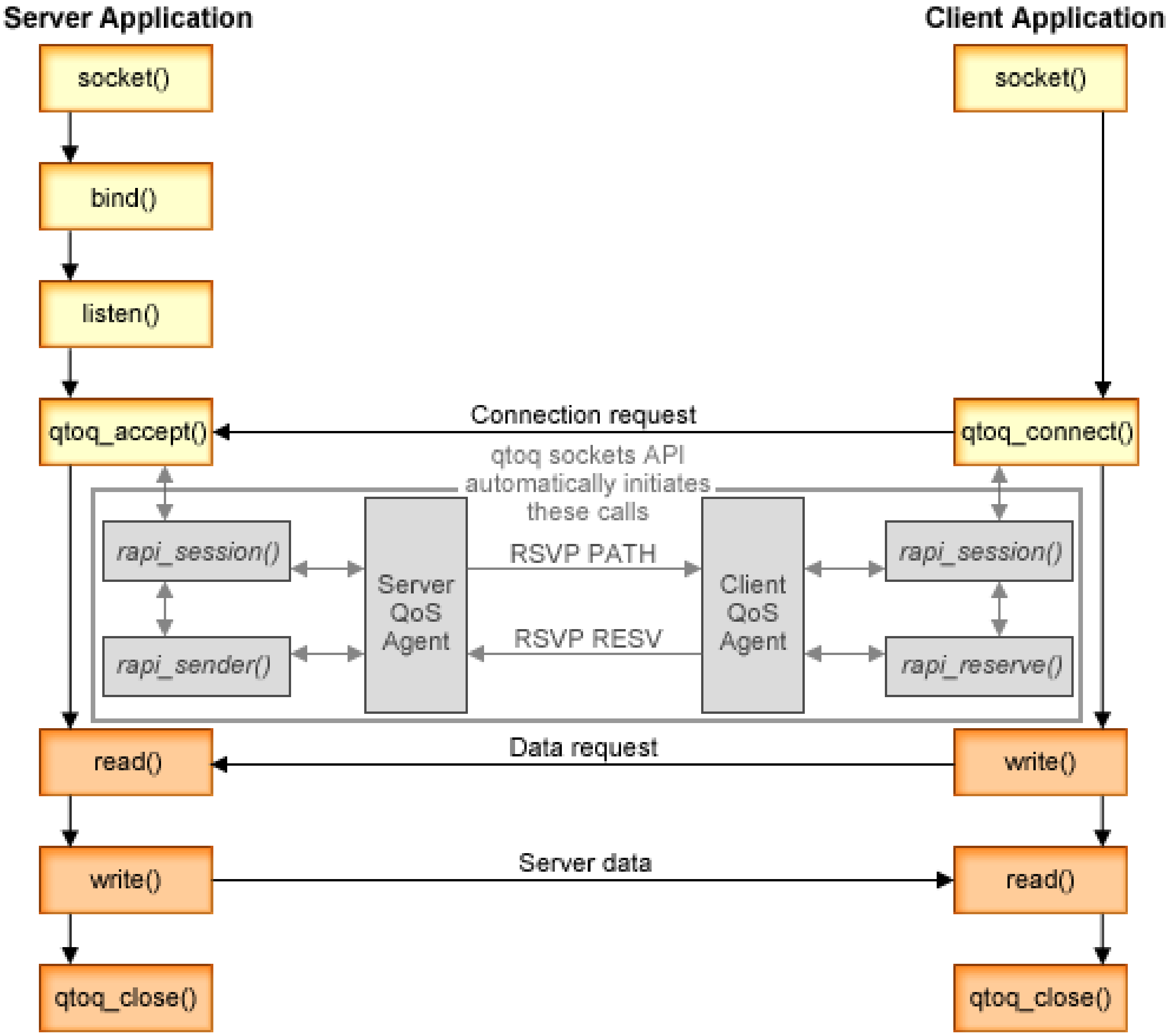
“Configure QoS with wizards” on page 51
To configure quality of service (QoS) policies, you must use the QoS wizards located in iSeries Navigator.

QoS API connection-oriented functional flow

The server and client examples in this topic illustrate qtoq quality of service (QoS) socket APIs written for a connection-oriented functional flow.

The following figure illustrates the client/server relationship of the qtoq socket functions enabled for QoS APIs 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 ReSerVation Protocol (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 figure. It also describes the relationship between the server and client application in a connection-oriented design. These are modifications of the basic socket APIs.

Server side

qtoq_accept() API for a rule marked with 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 is assigned.
5. The API calls the 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, and, if valid, the rule returns to the calling application with the results and the session ID.
7. The applications for the server and the client perform the required data transfers.
8. The application calls the qtoq_close() function to close the socket and to unload the rule.
9. The QoS server deletes the rule from the QoS manager, deletes the QoS session, and performs whatever other actions are needed.

qtoq_accept() API with normal RSVP 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. When a connection request comes in, the rapi_session() API is called to create a session with the QoS server for this connection and to get the QoS session ID, which is returned to the caller.
5. The rapi_sender() API is called to initiate a PATH message from the QoS server and to inform the QoS server that it must expect an 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, the server loads the appropriate rule with the QoS manager and sends a message to the application if the application requested notification on the call to the qtoq_accept() API.
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 deletes the rule from the QoS manager, deletes the QoS session, and performs whatever other actions are needed.

Client side

qtoq_connect() API with normal RSVP signaling

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 wants 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 is 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 responds by sending an RESV message to the QoS server on the application's server machine.
8. If the application requested notification, the QoS server sends the notification to the application by using 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 closes the QoS session and performs whatever other actions are necessary.

qtoq_connect() API for a rule marked with no signaling

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

Related reference

qtoq_accept() API

qtoq_close() API

rapi_session() API

rapi_sender() API

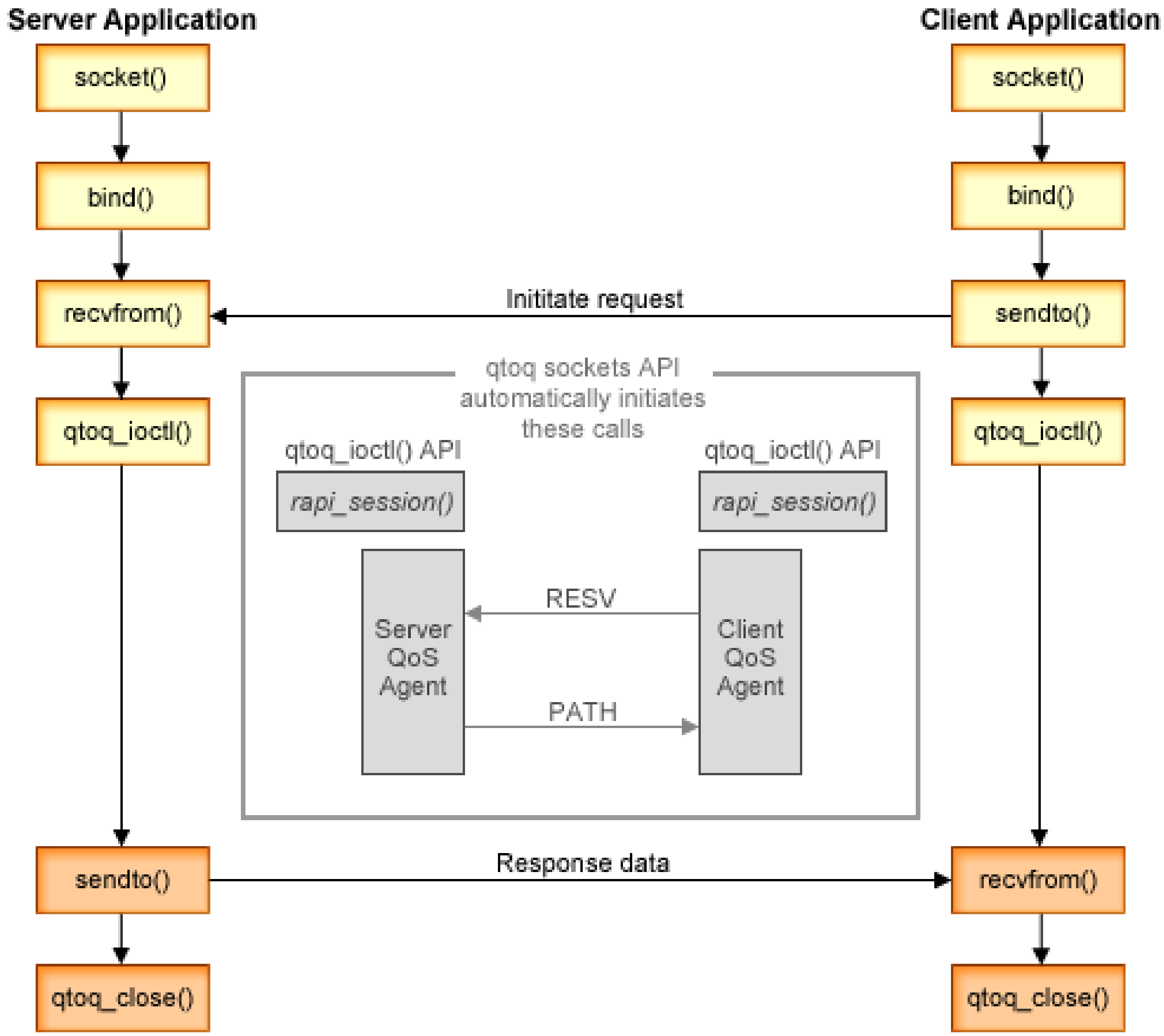
rapi_getfd() API

qtoq_connect() API

QoS API connectionless functional flow

The server and client examples in this topic illustrate qtoq quality of service (QoS) socket APIs written for a connectionless flow.

When the QoS-enabled API functions are called for a connectionless flow requesting that ReSerVation Protocol (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 figure. It also describes the relationship between the server and client application in a connectionless design. These are modifications of the basic socket APIs.

Server side

qtoq_ioctl() API for a rule marked with no signaling

1. qtoq_ioctl() API 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. The QoS server then 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 action is needed.

qtoq_ioctl() API with normal RSVP signaling

1. qtoq_ioctl() API sends message to the QoS server, requesting admission control for the requested connection.
2. The QoS server 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. It calls rapi_sender() to initiate a PATH message back to the client.
4. It then calls rapi_getfd() to get file descriptor in order to wait for QoS events.
5. The QoS server 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 other action is needed.

Client side

qtoq_ioctl() API with normal RSVP signaling

1. qtoq_ioctl() API 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. It 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 responds with the RESV message and then signal the application that the event has occurred through 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() API for a rule marked no-signaling

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

Related reference

qtoq_close() API

rapi_session() API

rapi_sender() API

rapi_getfd() API

qtoq_ioctl() API

QoS sendmsg() API extensions

The sendmsg() function is used to send data, ancillary data, or a combination of these through a connected or unconnected socket.

The sendmsg() API allows for quality of service (QoS) classification data. QoS policies use this function to define a more granular classification level for outgoing or incoming TCP/IP traffic. They specifically use ancillary data types that apply to the IP layer. The message type used is `IP_QOS_CLASSIFICATION_DATA`. This ancillary data can be used by the application to define attributes for traffic in a particular TCP connection. If the attributes passed by the application match the attributes defined in the QoS policy, then the TCP traffic is restricted by the policy.

Use the following information to initialize the `IP_QOS_CLASSIFICATION_DATA` structure:

- `ip_qos_version`: Indicates version of the structure. This must be filled in using the constant *IP_QOS_CURRENT_VERSION*.
- `ip_qos_classification_scope`: Specify a connection level scope (use constant *IP_QOS_CONNECTION_LEVEL*) or a message level scope (constant *IP_QOS_MESSAGE_LEVEL*).

  Connection level scope indicates that the QoS service-level obtained through classification of this message will remain in effect for all subsequent messages sent until the next sendmsg() with QoS classification data. Message level scope indicates that the QoS service level assigned will only be used for the message data included in this sendmsg() call. Future data sent without QoS classification data will inherit the previous connection level QoS assignment (from last Connection Level classification through sendmsg() or from the original TCP connection classification during connection establishment).
- `ip_qos_classification_type`: This specification indicates the type of classification data being passed. An application can chose to pass an application defined token, an application specified priority, or both a token and a priority. If the latter option is selected the two selected, classification types must be logically 'OR'ed. The following types can be specified:
  - Application defined token classification. A single type must be specified; if more than one is specified, the results are unpredictable.
    - `IP_SET_QOSLEVEL_W_APPL_TOKEN_ASCII`: This indicates that the classification data is a character string in ASCII format. When this option is specified, the application token needs to be passed in the ip_qos_appl_token field.

      Note: If the application needs to pass numeric values for the classification data it must first convert them to printable ASCII format. Also note that the string specified can be in mixed case and will be used in the exact format specified for comparison purposes.
    - `IP_SET_QOSLEVEL_W_APPL_TOKEN_EBCDIC`: Same as above except that the string is in EBCDIC format.

      Note: The `IP_SET_QOSLEVEL_W_APPL_TOKEN_ASCII` does perform slightly better than this option as the application data specified in the policy is saved in ASCII format inside of the TCP/IP stack, thereby eliminating the need to translate the application defined token on every sendmsg() request.
  - Application defined priority classification. A single type must be specified; if multiple priority types are specified, the results are unpredictable.
    - `IP_SET_QOSLEVEL_EXPIDITED`: Indicates that Expedited priority is requested
    - `IP_SET_QOSLEVEL_HIGH`: Indicates that High priority is requested
    - `IP_SET_QOSLEVEL_MEDIUM`: Indicates that Medium priority is requested
    - `IP_SET_QOSLEVEL_LOW`: Indicates that Low priority is requested
    - `IP_SET_QOSLEVEL_BEST_EFFORT`: Indicates that Best Effort priority is requested
  - `ip_qos_appl_token_len`: length of the ip_qos_appl_token specified.
  - `ip_qos_appl_token`: This "virtual field" immediately follows the ip_qos_classification_type field. The application classification token string in either ASCII or EBCDIC format depending on which flavor of IP_SET_QOSLEVEL_W_APPL_TOKEN_xxxx was specified for the classification type. This field is only referenced when an application defined token type is specified. Note that this string must not exceed 128 bytes. If a larger size is specified, only the first 128 bytes will be used. Also note that the length of the string is determined based on the value specified for cmsg_len (cmsg_len - sizeof(cmsghdr) - sizeof(ip_qos_classification_data)). This calculated length must not include any null terminating characters.

Related concepts

"Differentiated service" on page 2
This is the first type of outbound bandwidth policy you can create on your server. Differentiated service divides your traffic into classes. To carry out a differentiated service policy, you need to determine how you want to classify your network traffic and how to handle the different classes.

"Prioritized classes: How to classify network traffic" on page 3
Differentiated service identifies traffic as *classes*. The most common classes are defined using client IP addresses, application ports, server types, protocols, local IP addresses, and schedules. All traffic that conforms to the same class is treated equally.

Related reference

Sendmsg() API- Send a message over a socket

Directory server

You can export your policies to a directory server. Read this topic to see the advantages of using or not using a directory server, Lightweight Directory Access Protocol (LDAP) concepts and configuration, as well as the quality of service (QoS) schema.

QoS policy configuration can be exported to a directory server, using LDAP version 3.

Advantages to using a directory server

Exporting QoS policies to a directory server makes your policies easier to manage. There are three ways to use the directory server:

- The configuration data can be stored on one local directory server for many systems to share.
- The configuration data can be configured, stored, and only used by one system (not shared).
- The configuration data can reside on a directory server that holds data for other systems, but is not shared between those other systems. This allows you to use a single location to back up and save data for several systems.

Advantages to saving exclusively on your local server

Saving QoS policies on your local server is not as complex. There are a number of advantages to using policies locally:

- Eliminate the complexity of LDAP configuration for users who do not need it.
- Improve performance, because writing to LDAP is not the fastest method.
- Easier to duplicate a configuration between different iSeries. You can copy the file from one system to another. Because there is no primary or secondary machine, you can tailor each policy directly on the individual servers.

LDAP resources

If you decide to export your policies to an LDAP server, you must be familiar with LDAP concepts and directory structures before you continue. Within the QoS function in iSeries Navigator, you can configure a directory server that is used with your QoS policy.

Related concepts

IBM Directory Server for iSeries (LDAP)

"Configure directory server" on page 53
QoS policy configurations can be exported to an LDAP directory server.

Keywords

When you configure your directory server, you need to determine whether to associate keywords with each quality of service (QoS) configuration.

The keyword fields are optional and can be ignored. The following information helps explain the keyword concept and why you might want to use keywords.

In the QoS Initial Configuration wizard, you can configure a directory server. You can specify whether the server you configure is a primary system or a secondary system. The server that you maintain all your QoS policies on, 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 following descriptions 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 that 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.

Related concepts

"Distinguished name"
When you want to manage part of your directory, you refer to the *distinguished name (DN)* or (if you choose) a keyword.

"Configure directory server" on page 53
QoS policy configurations can be exported to an LDAP directory server.

Distinguished name

When you want to manage part of your directory, you refer to the *distinguished name (DN)* or (if you choose) a keyword.

You specify the DN when you configure the directory server within the quality of service (QoS) Initial Configuration wizard. DNs typically 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, say the LDAP server contains the directory structure as is shown in the following figure:

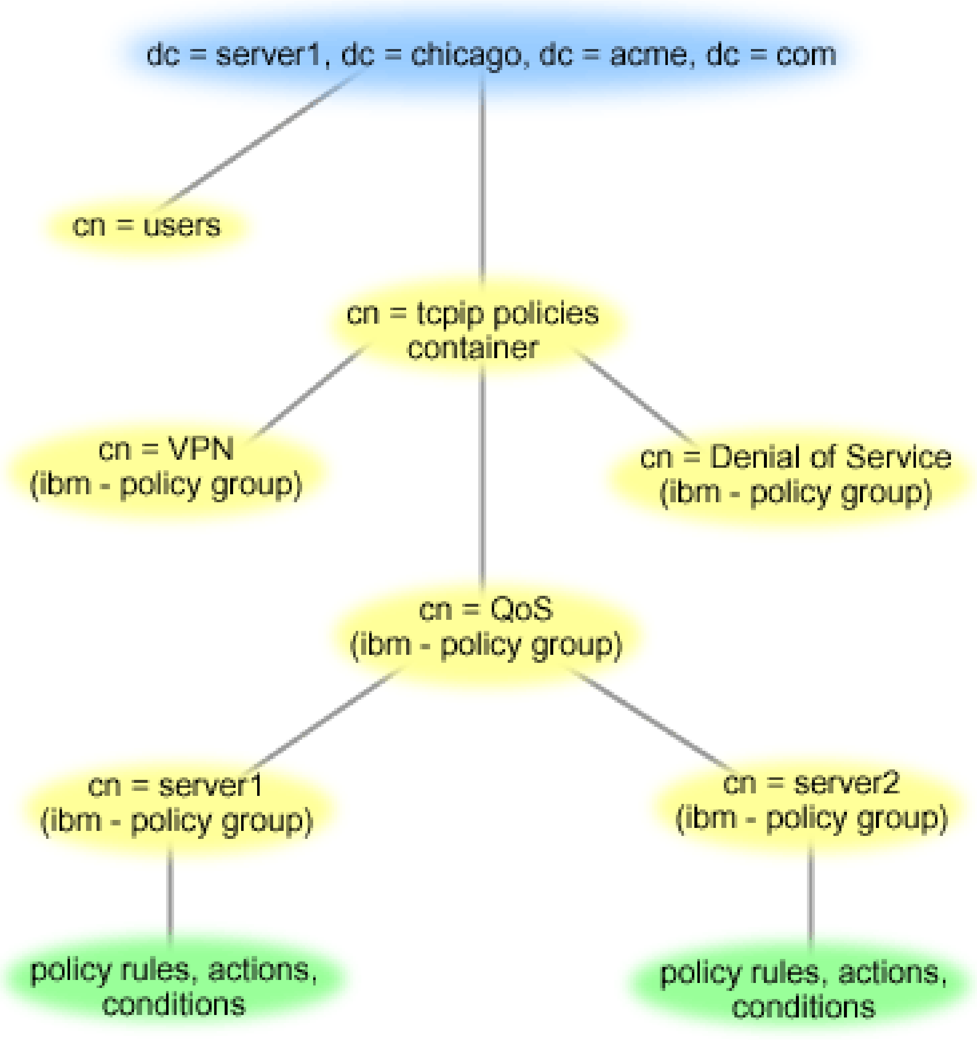

dc = server1, dc = chicago, dc = acme, dc = com
cn = users
cn = tcpip policies container
cn = VPN (ibm - policy group)
cn = Denial of Service (ibm - policy group)
cn = QoS (ibm - policy group)
cn = server1 (ibm - policy group)
cn = server2 (ibm - policy group)
policy rules, actions, conditions
policy rules, actions, conditions

Figure 3. 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 reads `cn=server1,cn=QoS,cn=tcpip policies,dc=server1,dc=chicago,dc=acme,dc=com`. On cn=server2, the default DN reads `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 because the string is typically too long to be displayed without scrolling.

Related concepts

"Keywords" on page 25
When you configure your directory server, you need to determine whether to associate keywords with each quality of service (QoS) configuration.

"Configure directory server" on page 53
QoS policy configurations can be exported to an LDAP directory server.

Related reference

"Related information for QoS" on page 67
Listed here are the IBM Redbooks (in PDF format), Web sites, and information center topics that relate to the quality of service (QoS) topic. You can view or print any of the PDFs.

Scenarios

These quality of service (QoS) policy scenarios can help you understand why and how to use QoS.

One of the best ways to learn about QoS is to see how the function works in your overall network picture. The following basic examples show why you need to use QoS policies and also provide some steps with instructions for creating the policies and classes of service.

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

Related concepts

"Monitor server transactions" on page 63
With the quality of service (QoS) monitor, you can verify that the QoS policies are working as you intend them to work.

Related reference

"Monitor QoS" on page 56
You can use the quality of service (QoS) monitor to analyze your IP traffic through the server.

Scenario: Limit browser traffic

You can use quality of service (QoS) to control traffic performance. Use a differentiated service policy to either limit or extend an application's performance within your network.

Situation

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.

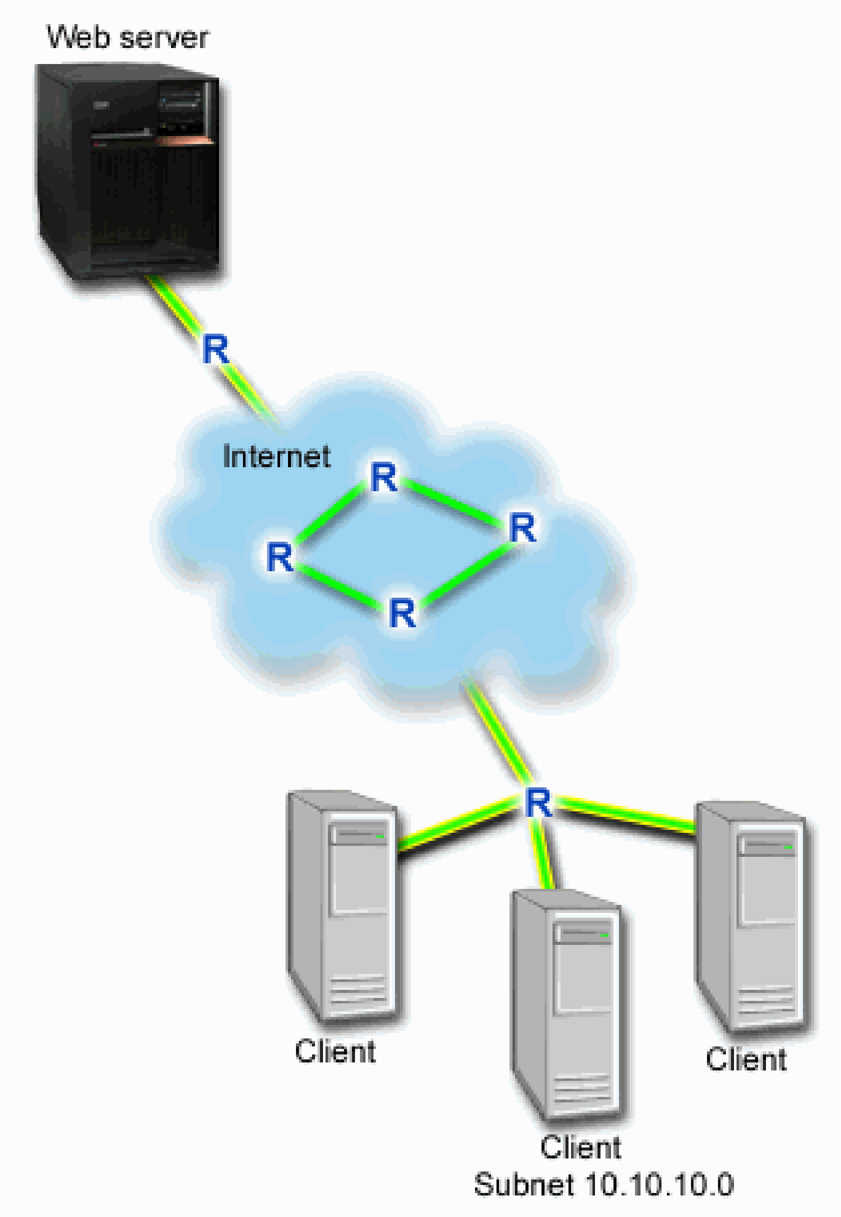


Figure 4. Web server limiting browser traffic to a client

Objectives

To limit browser traffic out of your network, you might 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 might be assigned a low codepoint value to affect how the network prioritizes browser traffic.

Prerequisites and assumptions

- You have a service-level agreement (SLA) with your ISP to ensure that the policies receive the requested priority. The QoS policy you create on the iSeries server enables traffic (in the policy) to receive priority throughout the network. The QoS policy does not guarantee the priority and is dependent on your SLA. In fact, taking advantage of QoS policies might give you some leverage to negotiate certain service-levels and rates.

- Differentiated service policies require DiffServ-aware routers along the network path. Most routers are DiffServ aware.

Configuration

After you verify the prerequisites steps, you are ready to create the differentiated service policy.

Related concepts

"Service-level agreement" on page 49
This topic points out some of the important aspects of a service-level agreement (SLA) that might affect your quality of service (QoS) implementation.

"Differentiated service" on page 2
This is the first type of outbound bandwidth policy you can create on your server. Differentiated service divides your traffic into classes. To carry out a differentiated service policy, you need to determine how you want to classify your network traffic and how to handle the different classes.

Related reference

"Monitor QoS" on page 56
You can use the quality of service (QoS) monitor to analyze your IP traffic through the server.

Scenario details: Create the differentiated service policy

1. In iSeries Navigator, expand iSeries A → **Network** → **IP Policies**.
2. Right-click **Quality of Service** and select **Configuration** to open the quality of service (QoS) interface.
3. On the QoS interface, right-click the DiffServ policy type and select **New Policy** to open the wizard.
4. Read the Welcome page and click **Next** to go to the Name page.
5. In the **Name** field, enter `UCD`. Optionally, you can also enter a description to help you remember the intent of this policy. Click **Next**.
6. On the Clients page, select **Specific address or addresses** and click **New** to define your client.
7. In the New Client window, enter the following information and click **OK**:
   - **Name**: `UCD_Client`
   - **IP address and mask**: `10.10.10.0 / 24`

   After you click OK, you return to the policy wizard. If you had previous clients created, de-select them and verify that only relevant clients are selected.
8. On the Server Data Request page, verify that **Any token** and **All priorities** are selected and click **Next**.
9. On the Applications page, select **Specific port, range of ports, or server type** and click **New**.
10. In the New Application window, enter the following information and click **OK** to return to the wizard:
    - **Name**: `HTTP`
    - **Port**: `80`
11. On the Applications page, select **Protocol** and verify **TCP** is selected. Click **Next**.
12. On the Local IP address page, verify **All IP addresses** is selected and click **Next**.
13. On the Differentiated Class of Service page, click **New** to define performance characteristics. The New Class of Service wizard appears.
14. Read the Welcome page and click **Next**.
15. On the Name page, enter `UCD_service`. Optionally, you can enter a description to help you remember the intent of this policy. Click **Next**.
16. On the Type of Service page, select **Outbound only** and click **Next**. This class of service will only be used for outbound policies.

17. On the Outbound DiffServ Codepoint Marking page, select **Class 4** and click **Next**. A per-hop behavior determines what performance this traffic will receive from routers and other servers on the network. Use the Help associated to the interface to assist in your decision.
18. On the Perform Outbound Traffic Metering page, verify that **Yes** is selected, and click **Next**.
19. On the Outbound Rate Control Limits page, enter the following information and click **Next**:
    - **Token bucket size**: 100 Kilobits
    - **Average rate limit**: 512 Kilobits per second
    - **Peak rate limit**: 1 Megabits per second
20. On the Outbound Out-of-Profile Traffic page, select **Drop UDP packets or reduce TCP congestion window** and click **Next**.
21. Review the Summary information for the class of service. If accurate, click **Finish** to create the class of service. After you click Finish, you return to the policy wizard and your class of service will be selected. Click Next.
22. On the Schedule page, select Active during selected schedule and click New.
23. In the Add New Schedule window, enter the following information and click **OK**:
    - **Name**: UCD_schedule
    - **Time of day**: Active 24 hours
    - **Day of week**: Friday
24. Click next to view a summary of the policy. If accurate, click **Finish**. On the QoS Server Configuration window, you can see the new policy listed in the right pane.

You are now finished configuring the differentiated service policy on iSeries A. The next step is to start or update the server.

Scenario details: Start or update the QoS server

On the quality of service (QoS) Server Configuration window, select **Server** → **Start** or **Server** → **Update**.

Scenario details: Use the monitor to verify your policy is working

To verify that the policy is working as you configured it to work, do the following steps:

1. On the quality of service (QoS) configuration window, select **Server** → **Monitor**. The QoS Monitor window appears.
2. Select the DiffServ policy type folder. This will display all the DiffServ policies. Select **UCD** from the list.

   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 indicate when traffic exceeds the configured policy values. In a differentiated service policy, the out-of-profile number (for UDP packets) indicates the number of bits being dropped. For TCP, the out-of-profile number indicates the number of bits that exceed the token bucket rate and are sent into the network. Bits are never dropped for TCP packets. The in-profile packets indicate the number of packets controlled by this policy (from the time the packet was started to the present monitor output).

   The value you assign to 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 "Monitor QoS" on page 56 for a description of all the monitor fields.

   Note: Remember that the results will only be accurate when the policy is active. Verify the schedule you specified within the policy.

Scenario details: Change properties (if needed)

After looking at the monitor results, you can change any policy or class of service properties to help achieve the results you expect.

You can change any of the values you created in the policy by doing the following steps:.

1. On the quality of service (QoS) Server Configuration window, select the **DiffServ** folder. Right-click **UCD** from the list in the right pane and select **Properties** to edit the policy. A Properties window appears with the values that control the general policy.
2. Change the appropriate values.
3. To edit the class of service, select the **Classes of service** folder. Right-click **UCD_service** from the list in the right pane and select **Properties** to edit the class of service. A QoS Properties window appears with the values that control traffic management.
4. Change the appropriate values.
5. From the QoS Server Configuration window, select **Server** → **Update** to accept your changes..

Scenario: Secure and predictable results (VPN and QoS)

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

Situation

You have a 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 figure 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.

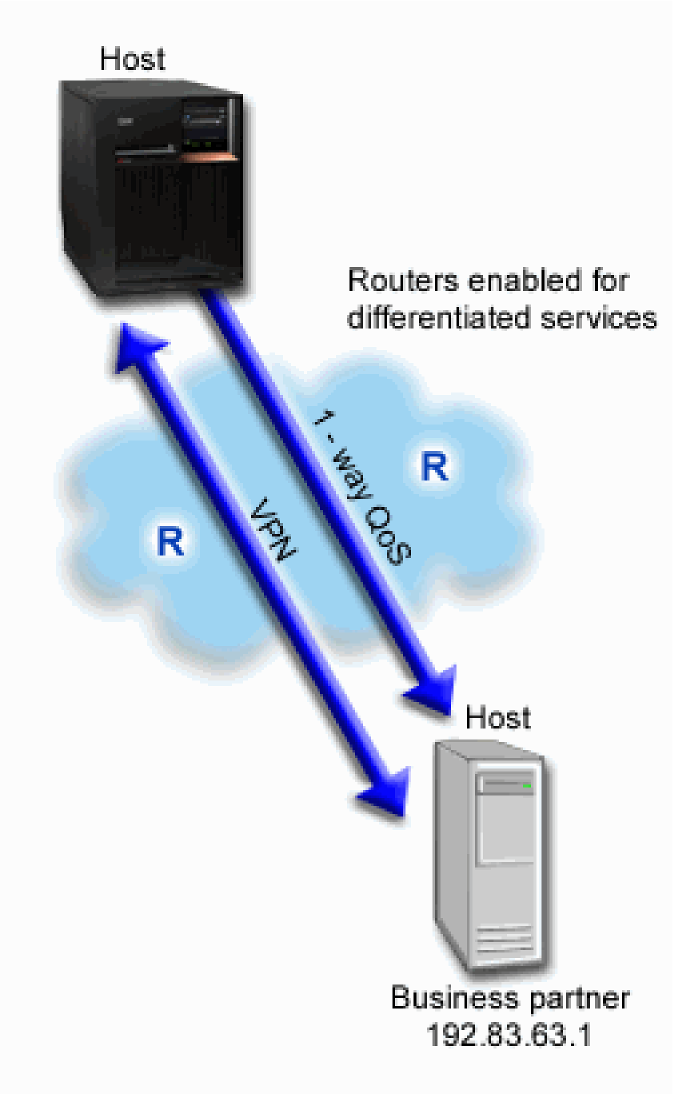


Figure 5. Host-to-host VPN connection using a QoS differentiated service policy

Objectives

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

Prerequisites and assumptions

- You have an SLA with your ISP to ensure that the policies receive the requested priority. The QoS policy you create on the iSeries server enables traffic (in the policy) to receive priority throughout the network. It does not guarantee it and is dependent on your SLA. In fact, taking advantage of QoS policies might give you some leverage to negotiate certain service-levels and rates. Use the service-level agreement link to find out more.
- Differentiated service policies require DiffServ-enabled routers along the network path. Most routers are DiffServ capable.

Configuration

After you verify the prerequisites steps, you are ready to create the differentiated service policy.

Related concepts

"Service-level agreement" on page 49
This topic points out some of the important aspects of a service-level agreement (SLA) that might affect your quality of service (QoS) implementation.

"Differentiated service" on page 2
This is the first type of outbound bandwidth policy you can create on your server. Differentiated service divides your traffic into classes. To carry out a differentiated service policy, you need to determine how you want to classify your network traffic and how to handle the different classes.

Related reference

"Monitor QoS" on page 56
You can use the quality of service (QoS) monitor to analyze your IP traffic through the server.

Scenario details: Set up a host-to-host VPN connection

See the Host-to-Host VPN connection example, to assist you with the VPN configuration.

Scenario details: Create the differentiated service policy

1. In iSeries Navigator, expand iSeries A → **Network** → **IP Policies**.
2. Right-click **Quality of Service** and select **Configuration** to open the quality of service (QoS) Server Configuration window.
3. On the QoS Server Configuration window, right-click DiffServ and select **New Policy** to open the wizard.
4. Read the Welcome page and click **Next** to go to the **Name** page.
5. In the **Name** field, enter `VPN` and click **Next**. Optionally, you can enter a description to help you remember the intent of this policy.
6. On the Clients page, select **Specific address or addresses** and click **New** to define your client.
7. On the New client window, enter the following information:
   - **Name**: VPN_Client
   - **IP address**: 192.83.63.1
   - Click **OK** to create the client and return to the differentiated service wizard.

   After you click OK, you return to the policy wizard. If you have previous clients created, de-select them and verify that only relevant clients are selected.
8. On the Server Data Request page, verify that **Any token** and **All priorities** are selected.
9. On the Applications page, verify that **All ports** and **All** are selected.
10. Click **Next**.
11. On the Local IP address page, accept the default value and click **Next**.
12. On the Differentiated Class of Service page, click **New** to define performance characteristics. The New Class of Service wizard appears.
13. Read the Welcome page and click **Next**.
14. On the Name page, enter `EF_VPN`.
15. On the Type of Service page, select **Outbound only** and click **Next**. This class of service will only be used for outbound policies.
16. On the Outbound DiffServ Codepoint Marking page, select **Class 3**. A per-hop behavior determines what performance this traffic will receive from routers and other servers on the network. Use the Help associated to the interface to assist in your decision.
17. On the Perform Outbound Traffic Metering page, verify that **Yes** is selected and click **Next**.
18. On the Outbound Rate Control Limits page, enter the following information and click **Next**:

- **Token bucket size**: 100 Kilobits
- **Average rate limit**: 64 Megabits per second
- **Peak rate limit**: Do not limit

19. On the Outbound Out-of-Profile Traffic page, select **Drop UDP packets or reduce TCP congestion window** and click **Next**.
20. Review the Class of Service summary page and click **Finish** to return to the policy wizard.
21. On the Differentiated Class of Service page, verify that **EF_VPN** is selected and click **Next**.
22. On the Schedule page, select **Active during selected schedule** and click **New**.
23. On the Add New Schedule window, enter the following information and click **OK**:
    - **Name**: FirstShift
    - **Time of day**: Active at specific times and add 9:00 a.m. to 5:00 p.m.
    - **Day of week**: Active on specific day and select Monday through Friday.
24. On the Schedule page, click **Next**.
25. Review the Summary information. If accurate, click **Finish** to create the policy. The QoS Server Configuration window lists all the policies created on the server. After you complete the wizard, the policy is listed in the right pane.

You are now finished configuring the differentiated service policy on iSeries A. The next step is to start or update the server.

Scenario details: Start or update the QoS server

On the quality of service (QoS) Server Configuration window, select **Server** → **Start** or **Server** → **Update**.

Scenario details: Use the monitor to verify your policy is working

To verify that the policy is working as you configured it to work, do the following steps:

1. On the quality of service (QoS) Server Configuration window, select **Server** → **Monitor**. The QoS Monitor window appears.
2. Select the DiffServ policy type. This will display all the DiffServ policies.

   Similar to example 1, the most interesting fields are the fields that obtain their data from your traffic. These fields include the bits total, bits in-profile, and packets out-of-profile fields. Bits out-of-profile indicate when traffic exceeds the configured policy values. The in-profile packets indicate the number of packets controlled by this policy. What values you assign the average rate limit field is very important. When TCP packets exceed this limit, they are sent into the network, until the TCP congestion window can be reduced to queue out-of-profile packets. As a result, the bits out-of-profile will increase. The difference between this policy and the Limit browser traffic scenario is that the packets here are protected using the VPN protocol. As you can see, QoS does work with a VPN connection. See the “Monitor QoS” on page 56 for a description of all the monitor fields.

   Note: Remember that the results will only be accurate when the policy is active. Verify the schedule you specified within the policy.

Scenario details: Change properties (if needed)

After looking at the monitor results, you can change any policy or class of service properties to help achieve the results you expect.

To edit the class of service after you create it, do the following steps:

1. On the quality of service (QoS) Server Configuration window, select the **DiffServ** folder. Right-click **VPN** from the list in the right pane and select **Properties** to edit the policy. A Properties dialog box appears with the values that control the general policy.

2. Change the appropriate values.
3. To edit the class of service, select the **Classes of service** folder. Right-click **EF_VPN** from the list in the right pane and select **Properties** to edit the class of service. A QoS Properties dialog box appears with the values that control traffic management.
4. Change the appropriate values.
5. From the QoS Server Configuration window, select **Server** → **Update** to accept your changes.

Scenario: Limit inbound connections

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

Situation

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 on the local interface 192.168.1.1. Quality of service (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 do an inbound admission policy, which will restrict the number of accepted inbound connections.

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

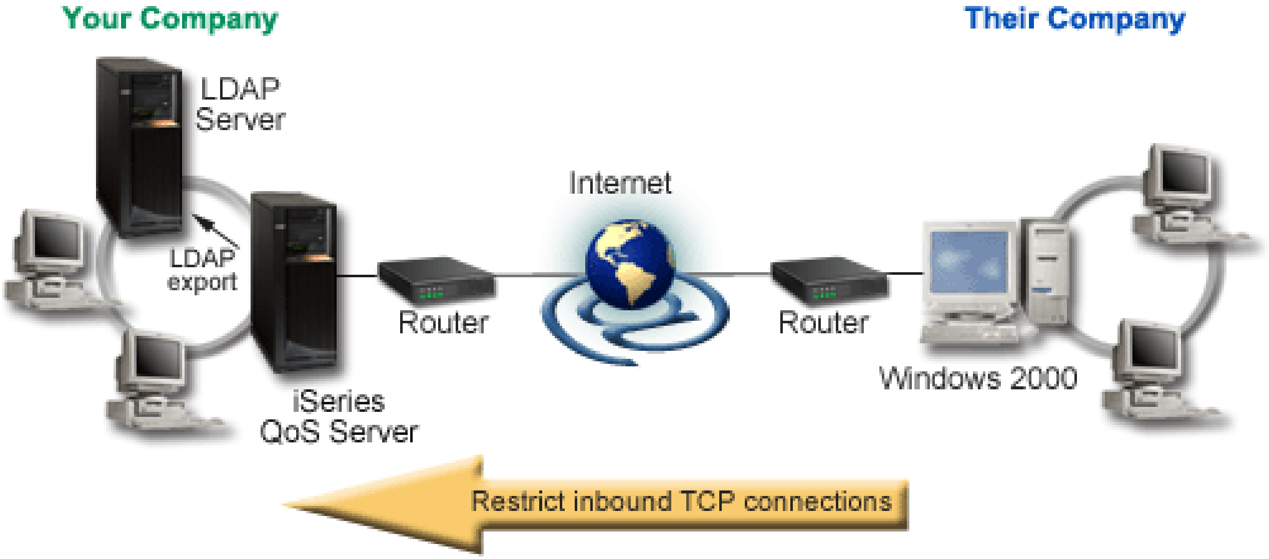


Figure 6. Restricting inbound TCP connections

Objectives

To configure an inbound policy, you must decide whether you are restricting traffic to a local interface or a specific application and whether 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 192.168.1.1.

Configuration

These topics show how to create an inbound admission policy.

Related reference

"Monitor QoS" on page 56
You can use the quality of service (QoS) monitor to analyze your IP traffic through the server.

Scenario details: Create the inbound admission policy

1. In iSeries Navigator, expand iSeries A → **Network** → **IP Policies**.
2. Right-click **Quality of Service** and select **Configuration** to open the quality of service (QoS) Server Configuration window.
3. On the QoS Server Configuration window, right-click **Inbound Admission Policies** and select **New Policy** to open the wizard.
4. Read the Welcome page and click **Next**.
5. In the **Name** field, enter `Restrict_TheirCo` and click **Next**. Optionally, you can enter a description to help you remember the intent of this policy.
6. On the Clients page, select **Specific address or addresses** and click **New** to define your client.
7. On the New client window, enter the following information:
   - **Name**: Their_Co
   - **IP address range**: 10.1.1.1 to 10.1.1.10
   - Click **OK** to create the client and return to the policy wizard.

   After you click OK, you return to the policy wizard. If you had previous clients created, de-select them and verify that only relevant clients are selected.
8. On the Uniform Resource Identifier (URI) page, verify **Any URI** is selected and click **Next**.
9. On the Applications page, select **Specific port, range of ports, or server type** and click **New**.
10. On the New Application window, enter the following information and click **OK** to return to the wizard:
    - **Name**: HTTP
    - **Port**: 80
11. Click **Next** to go the Codepoint page.
12. On the Codepoint page, verify **All codepoints** is selected and click **Next**.
13. On the Local IP Address page, select **IP address** and select an interface on which requests are made to your local system. In this example, use 192.168.1.1.
14. On the Class of Service page, click **New** to define performance characteristics. The New Class of Service wizard appears.
15. Read the Welcome page and click **Next**.
16. On the Name page, enter **inbound** and click **Next**. Optionally, you can add a description to help you remember the intent of this class of service.
17. On the Type of Service page, select **Inbound only**. This class of service will only be used for inbound policies.
18. On the Inbound Limits page, enter the following information and click **Next**:
    - Average connection rate: 50 per second
    - Connection burst limit: 50 connections
    - Priority: Medium
19. Click **Finish** to return to the policy wizard.
20. On the Class of service page, verify that the class of service you just created is selected and click **Next**.
21. On the Schedule page, select **Active during selected schedule** and click **New**.
22. On the New Schedule window, enter the following information and click **OK**:
    - Name: FirstShift
    - Time of day: Active at specific times and add 9:00 to 5:00.

- Day of week: Active on specific days and select Monday through Friday.

23. On the Schedules page, click **Next**.
24. Review the Summary information. If accurate, click **Finish** to create the policy. The QoS Server Configuration lists all the policies created on the server. After you complete the wizard, the policy is listed in the right pane.

    You are now finished configuring the Inbound Admission policy on iSeries A. The next step is to start or update the server.

Scenario details: Start or update the QoS server

On the quality of service (QoS) Server Configuration window, select **Server** → **Start** or **Server** → **Update**.

Scenario details: Use the monitor to verify your policy is working

To verify that the policy is working as you configured it to work, do the following steps:

1. On the quality of service (QoS) configuration window, select **Server** → **Monitor**. The QoS Monitor window appears.
2. Select Inbound admission policy type. This will display all the Inbound admission policies. Select **Restrict_TheirCo** from the list.

   Make sure to check any measured fields, such as accepted requests, dropped requests, total requests, and connection rate. Dropped requests 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 to 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 "Monitor QoS" on page 56 section for a description of all the monitor fields.

   Note: Remember that the results will only be accurate when the policy is active. Verify the schedule you specified within the policy.

Scenario details: Change properties (if needed)

After looking at the monitor results, you can change any policy or class of service properties to help achieve the results you expect.

1. On the quality of service (QoS) Server Configuration window, select the **Inbound admission** folder. Right-click **Restrict_TheirCo** from the list in the right pane and select **Properties** to edit the policy. A Properties window appears with the values that control the general policy.
2. Change the appropriate values.
3. To edit the class of service, select the **Classes of service** folder. Right-click **inbound** from the list in the right pane and select **Properties** to edit the class of service. A QoS Properties window appears with the values that control traffic management.
4. Change the appropriate values.
5. From the QoS Server Configuration window, select **Server** → **Update** to accept your changes.

Scenario: Predictable B2B traffic

If you need predictable delivery and still need to request a reservation, you also use an integrated service policy. This example uses a controlled load service.

Situation

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 on demand business service. You need to provide predictable transactions to your customers. You want to give the sales unit a higher quality of service (QoS) for their ordering application during the busiest time of the day (between 10:00 a.m. and 4:00 p.m.).

In the following figure, the sales team is within your private network. There are routers, enabled by ReSerVation Protocol (RSVP), along the traffic's path to the B2B client. Each `R` represents a router along the traffic's path.

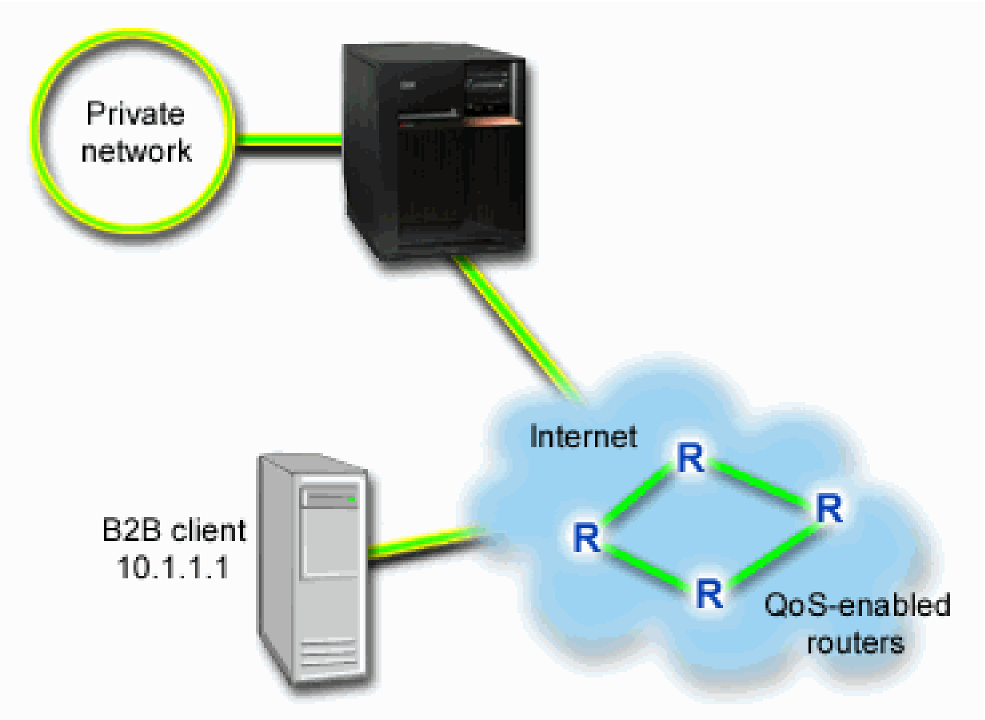


Figure 7. Integrated services policy to a B2B client using RSVP-enabled routers

Objectives

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. Because this particular application is tolerant to some delay, you decide to use an integrated services policy using a controlled load service.

Integrated service policies also require that, along the traffic's path, the routers are RSVP-enabled.

Prerequisites and assumptions

An integrated service policy is an advanced policy that can require substantial resource. Integrated service policies require the following prerequisites:

- **RSVP-enabled applications**

  Because your server does not have any RSVP-enabled applications, you must write your own RSVP-enabled applications. To write your own applications, use the RSVP API, qtoq QoS socket APIs, or integrated service APIs.
- **RSVP-enabled routers and servers along the network path**

QoS is a network solution. If you are unsure if the entire network has RSVP capabilities, you can still create an integrated service policy and use a marking to give it some priority; however, priority cannot be guaranteed.

- **Service-level agreement**

  You have a service-level agreement (SLA) with your ISP to ensure that the policies receive the requested priority. The QoS policy you create on the iSeries server enables traffic (in the policy) to receive priority throughout the network. The QoS policy does not guarantee the priority and is dependent on your SLA. In fact, taking advantage of QoS policies can give you some leverage to negotiate certain service-levels and rates. Use the service-level agreement link to find out more.

  Note: If you are within a private network, an SLA is not required.

Configuration

After you verify the prerequisites steps, you are ready to create the integrated service policy.

Related concepts

"Integrated service types" on page 9
There are two integrated service types: controlled load and guaranteed.

"Integrated service" on page 6
The second type of outbound bandwidth policy you can create is an integrated service policy. Integrated service provides the capability for IP applications to request and reserve bandwidth using the ReSerVation Protocol (RSVP) and quality of service (QoS) APIs.

"QoS APIs" on page 16
You can read this topic to learn about protocols, APIs, and requirements for a router that is enabled for the ReSerVation Protocol (RSVP). The current quality of service (QoS) APIs include the RAPI API, the qtoq socket API, the sendmsg() API, and the monitor APIs.

"Service-level agreement" on page 49
This topic points out some of the important aspects of a service-level agreement (SLA) that might affect your quality of service (QoS) implementation.

Related reference

"Monitor QoS" on page 56
You can use the quality of service (QoS) monitor to analyze your IP traffic through the server.

Scenario details: Create the integrated service policy

1. In iSeries Navigator, expand iSeries A → **Network** → **IP Policies**.
2. Right-click **Quality of Service** and select **Configuration** to open the quality of service (QoS) Server Configuration window.
3. On the QoS Server Configuration window, right-click the IntServ policy type and select **New Policy** to open the wizard.
4. Read the Welcome page and click **Next** to go to the **Name** page.
5. In the **Name** field, enter `B2B_CL` and click **Next**. Optionally, you can enter a description to help you remember the intent of this policy.
6. On the Clients page, select **Specific address or addresses** and click **New** to define your client.
7. On the New Client window, enter the following information:
   - **Name**: CL_client
   - **IP address**: 10.1.1.1
   - Click **OK** to create the client and return to the policy wizard.

   After you click OK, you return to the policy wizard. If you have previous clients created, de-select them and verify that only relevant clients are selected.
8. On the New Application window, enter the following information and click **OK** to return to the wizard:

- **Name**: business_app
- **Port range**: 7000-8000

9. On the Applications page, select **Protocol** and verify that **TCP** is selected. Click **Next**.

   Note: The application you select for an integrated service policy must be written to use the Resource Reservation Setup Protocol (RAPI) API or the qtoq sockets APIs. Along with the ReSerVation Protocol (RSVP), these APIs perform the integrated service reservation through the network. If you do not utilize these APIs, the application will not receive any priority or guarantee. It is also important to note that this policy enables your applications to receive priority through a network, but cannot guarantee it. All routers and servers along the traffic's path must also use the RSVP protocol to guarantee a reservation. An end-to-end reservation is dependent on participation throughout the network.
10. On the Local IP address page, accept the default value and click **Next**.
11. On the Integrated Services Type page, select **Controlled load** and click **Next**.
12. On the Integrated Services Marking page, select **No, do not assign a per-hop behavior** and click **Next**.
13. On the Integrated Services Performance Limits page, enter the following information and click **Next**:
    - **Maximum number of flows**: 5
    - **Token rate limit (R)**: Do not limit
    - **Token bucket size**: 100 Kilobits
    - **Token rate limit (R)**: 25 Megabits per second
14. On the Schedule page, select **Active during selected schedule** and click **New**.
15. On the New Schedule page, enter the following information and click **OK**:
    - **Name**: primetime
    - **Time of day**: Active at specific times and add 10:00 a.m. to 4:00 p.m.
    - **Day of week**: Active on specific day and select Monday through Friday.
16. On the Schedules page, click **Next**.
17. Review the Summary information. If accurate, click **Finish** to create the policy. The main QoS interface lists all the policies created on the server. After you complete the wizard, the policy is listed in the right pane.

You are now finished configuring the integrated service policy on iSeries A. The next step is to start or update the server.

Scenario details: Start or update the QoS server

On the quality of service (QoS) Server Configuration window, select **Server** → **Start** or **Server** → **Update**.

Scenario details: Use the monitor to verify your policy is working

To verify that the policy is working as you configured it to work, do the following steps:

1. On the quality of service (QoS) Server Configuration window, select **Server** → **Monitor**. The QoS Monitor window appears.
2. Select the IntServ policy type. This will display all the IntServ policies.

   The most interesting fields are the fields that obtain their data from your traffic. Make sure to check the bits total, bits in-profile, and packets in-profile fields. Bits out-of-profile 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 QoS" on page 56 section.

Note: Remember that the results will only be accurate when the policy is active. Verify the schedule you specified within the policy. Also, the monitor only shows IntServ policies after the applications are running. A ReSerVation Protocol (RSVP) reservation has to be established before monitoring.

Scenario details: Change properties (if needed)

After looking at the monitor results, you can change any policy properties to help achieve the results you expect.

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

1. On the quality of service (QoS) Server Configuration window, select the **IntServ** folder. Right-click **B2B_CL** from the list in the right pane and select **Properties** to edit the policy. A Properties window appears with the values that control the general policy.
2. Change the appropriate values.
3. From the QoS Server Configuration window, select **Server** → **Update** to accept your changes.

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, guaranteed service is used.

Situation

The chief executive officer (CEO) of your company is going to give a live broadcast to a client across the country between 1:00 p.m.- 2:00 p.m. You must guarantee that IP telephony will have guaranteed bandwidth so that there are no interruptions during the broadcast. In this scenario, the application resides on the server.

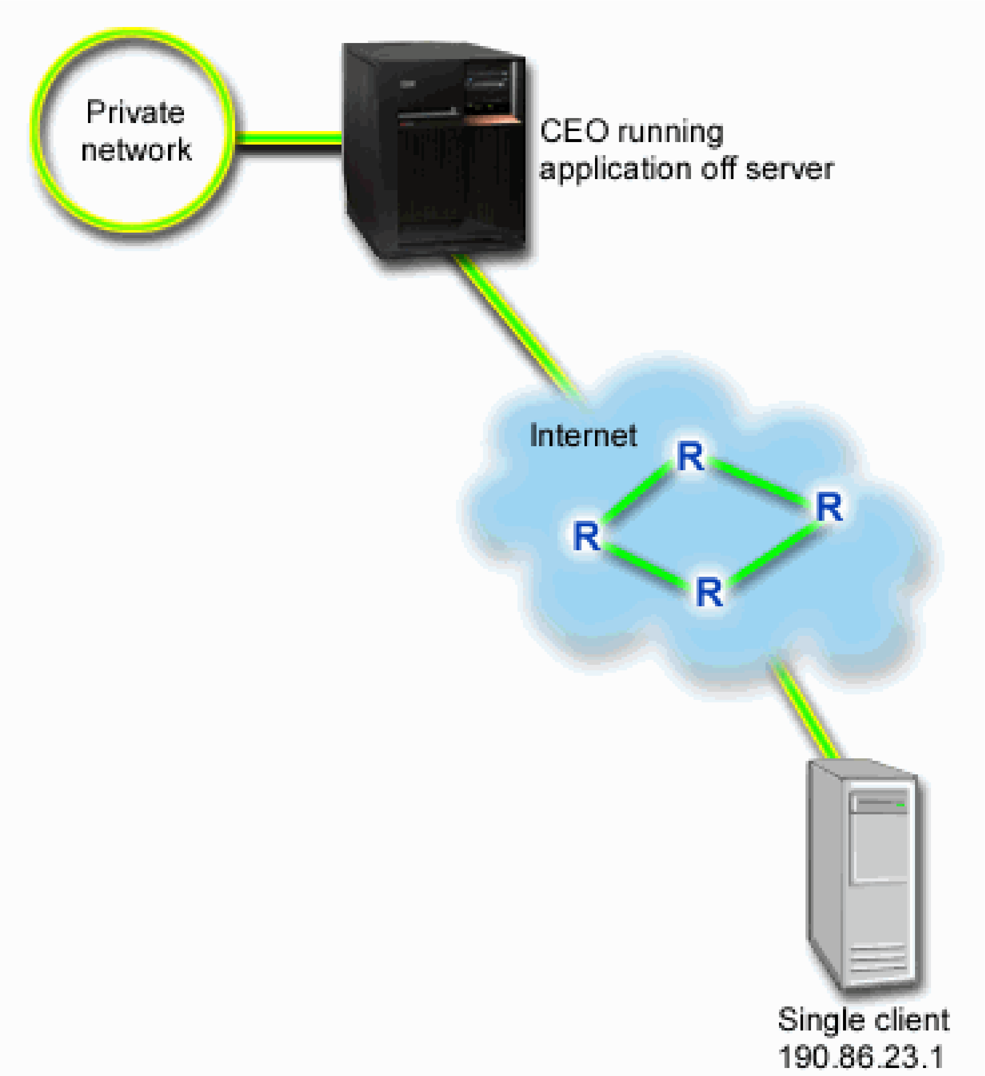


Figure 8. CEO to client presentation guaranteed by an integrated service policy

Objectives

Because 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.

Prerequisites and assumptions

An integrated service policy is an advanced policy that can require substantial resource. Integrated service policies require the following prerequisites:

- **RSVP-enabled applications**

  Because your server does not have any RSVP-enabled applications, you must write your own RSVP-enabled applications. To write your own applications, use the ReSerVation Protocol (RAPI) API or qtoq quality of service (QoS) socket APIs. For more information, see “QoS APIs” on page 16 and look for integrated service APIs.
- **RSVP-enabled routers and servers along the network path**

QoS is a network solution. If you are unsure if the entire network has RSVP capabilities, you can still create an integrated service policy and use a marking to give it some priority; however, priority cannot be guaranteed.

- **Service-level agreement**

  You have a service-level agreement (SLA) with your ISP to ensure that the policies receive the requested priority. The QoS policy you create on the iSeries server enables traffic (in the policy) to receive priority throughout the network. The QoS policy does not guarantee the priority and is dependent on your SLA. In fact, taking advantage of QoS policies can give you some leverage to negotiate certain service-levels and rates. Use the service-level agreement link to find out more.

Configuration

After you verify the prerequisites steps, you are ready to create the integrated service policy.

Related concepts

"Integrated service types" on page 9
There are two integrated service types: controlled load and guaranteed.

"Integrated service" on page 6
The second type of outbound bandwidth policy you can create is an integrated service policy. Integrated service provides the capability for IP applications to request and reserve bandwidth using the ReSerVation Protocol (RSVP) and quality of service (QoS) APIs.

"Service-level agreement" on page 49
This topic points out some of the important aspects of a service-level agreement (SLA) that might affect your quality of service (QoS) implementation.

Related reference

"Monitor QoS" on page 56
You can use the quality of service (QoS) monitor to analyze your IP traffic through the server.

Scenario details: Create the integrated service policy

1. In iSeries Navigator, expand iSeries A → **Network** → **IP Policies**.
2. Right-click **Quality of Service** and select **Configuration** to open the quality of service (QoS) Server Configuration window.
3. On the QoS Server Configuration window, right-click the IntServ policy type and select **New Policy** to open the wizard.
4. Read the Welcome page and click **Next** to go to the **Name** page.
5. In the **Name** field, enter `CEO_guaranteed` and click **Next**. Optionally, you can enter a description to help you remember the intent of this policy.
6. On the Clients page, select **Specific address or addresses** and click **New** to define your client.
7. On the New Client window, enter the following information:
   - **Name**: Branch1
   - **IP address**: 190.86.23.1
   - Click **OK** to create the client and return to the integrated service wizard.

   After you click OK, you return to the policy wizard. If you have previous clients created, de-select them and verify that only relevant clients are selected. On the Applications page, select **Specific port, range of ports, or server type** and click **New**.
8. On the New Application window, enter the following information and click **OK** to return to the wizard:
   - **Name**: IP telephony
   - **Port**: 2427
9. On the Applications page, select **Protocol** and verify that **TCP** is selected. Click **Next**.

Note: The application you select for an integrated service policy must be written to use the Resource Reservation Setup Protocol (RAPI) API or the qtoq sockets APIs. Along with the ReSerVation Protocol (RSVP), these APIs perform the integrated service reservation through the network. If you do not utilize these APIs, the application will not receive any prioritization or guarantee. It is also important to note that this policy enables your applications to receive priority through a network, but cannot guarantee it. All routers and servers along the traffic's path must also use the RSVP protocol to guarantee a reservation. An end-to-end reservation is dependent on participation throughout the network.

10. On the Local IP address page, accept the default value, **All IP addresses**.
11. On the Integrated Services Type page, select **Guaranteed** and click **Next**.
12. On the Integrated Services Marking page, select **No, do not assign a per-hop behavior** and click **Next**.
13. On the Integrated Services Performance Limits page, enter the following information and click **Next**:
    - **Maximum number of flows**: 1
    - **Aggregate bandwidth limit (R)**: Do not limit
    - **Token bucket size**: 100 Kilobits
    - **Bandwidth limit (R)**: 16 Megabits per second
14. On the Schedule page, select **Active during selected schedule** and click **New**.
15. On the New Schedule page, enter the following information and click **OK**:
    - **Name**: one_hour
    - **Time of day**: Active at specific times and add 1:00 p.m. to 2:00 p.m.
    - **Day of week**: Active on specific day and select Monday.
16. On the Schedule page, click **Next**.
17. Review the Summary information. If accurate, click **Finish** to create the policy. The main QoS Server Configuration window lists all the policies created on the server. After you complete the wizard, the policy is listed in the right pane.

You are now finished configuring the integrated service policy on iSeries A. The next step is to start or update the server.

Scenario details: Start or update the QoS server

On the quality of service (QoS) Server Configuration window, select **Server** → **Start** or **Server** → **Update**.

Scenario details: Use the monitor to verify your policy is working

To verify that the policy is working as you configured it to work, do the following steps:

1. On the quality of service (QoS) Server Configuration window, select **Server** → **Monitor**. The QoS Monitor window appears.
2. Select the IntServ policy type folder. This will display all the IntServ policies.

   The most interesting fields are the measured fields that obtain their data from your traffic. These fields include the bits total, bits in-profile, and packets in-profile. Bits out-of-profile indicates that other traffic is getting delayed or dropped to satisfy this integrated service policy requirements. See the "Monitor QoS" on page 56 section for a description of all the monitor fields.

   Note: Remember that the results will only be accurate when the policy is active. Verify the schedule you specified within the policy. Also, the monitor only shows IntServ policies after the applications are running. A ReSerVation Protocol (RSVP) reservation has to be established before monitoring.

Scenario details: Change properties (if needed)

After looking at the monitor results, you can change any policy properties to help achieve the results you expect.

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

1. On the quality of service (QoS) Server Configuration window, select the **IntServ** folder. Right-click **CEO_guaranteed** from the list in the right pane and select **Properties** to edit the policy. A Properties window appears with the values that control the general policy.
2. Change the appropriate values.
3. From the QoS Server Configuration window, select **Server** → **Update** to accept your changes.

Scenario: Monitor current network statistics

Within the wizards, you are asked to set performance limits. These values cannot be recommended because they are based on individual network requirements.

Objectives

To set these limits, you really need to understand your current network performance. Because you are trying to configure quality of service (QoS) policies, you probably already have a good idea of your current network needs. To determine exact rate limits, such as token bucket rate, you might want to monitor all the traffic on your server so that 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.

Related concepts

"Token bucket and bandwidth limits" on page 10
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.

"Average connection rate and burst limits" on page 16
Connection rates and burst limits, together, are known as *rate limits*. These rate limits help restrict inbound connections trying to enter your server. Rate limits are set in a class of service used with inbound admission policies.

Related reference

"Monitor QoS" on page 56
You can use the quality of service (QoS) monitor to analyze your IP traffic through the server.

Scenario details: 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 quality of service (QoS) policy wizard appears.

Scenario details: Create a differentiated service policy

Because 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.

Scenario details: Complete a new class of service

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

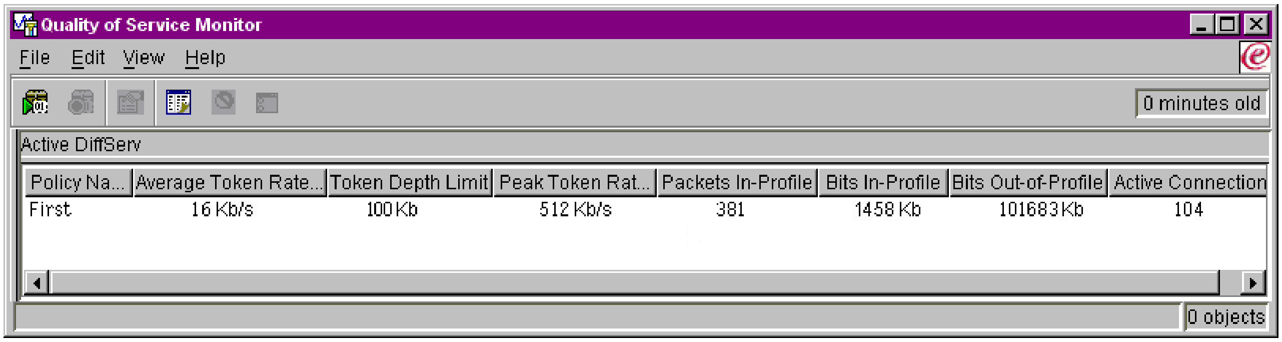
Classes of service actually determine the performance levels that this traffic receives from a router. You might name your class of service **Unlimited**, to show that this traffic receives a higher service. iSeries Navigator lists all the classes of service defined on your server.

Scenario details: 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, Inbound admission).
2. Right-click the policy that you want to monitor and select **Monitor**.

The following figure is a list of possible monitor output for the policy set above.



Policy Na...	Average Token Rate...	Token Depth Limit	Peak Token Rat...	Packets In-Profile	Bits In-Profile	Bits Out-of-Profile	Active Connection
First	16 Kb/s	100 Kb	512 Kb/s	381	1458 Kb	101683 Kb	104

Figure 9. Quality of service (QoS) 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 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 to 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. Monitor QoS provides full descriptions of all the monitor fields.

Scenario details: Change values when needed

After you monitor, you can change any of the values you previously selected. Right-click the class of service name you created in this policy. When you select **Properties**, a QoS Properties window appears with the values that control your traffic.

Scenario details: Monitor the policy again

After seeing the results, use the ″guess and check″ method to find the best limits for your network needs.

Plan for QoS

The most important step to accomplishing quality of service (QoS) is planning. To receive expected results, you must review your network equipment and monitor network traffic.

This topic also includes a planning advisor. 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 configuring QoS.

Consider 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 carry out any policies, you might want to use the QoS monitor to verify your IP traffic's current performance levels. These results can help you determine where congestion is occurring.

Related concepts

"Monitor server transactions" on page 63
With the quality of service (QoS) monitor, you can verify that the QoS policies are working as you intend them to work.

"Configure QoS" on page 51
You can use this topic to create differentiated service policies, integrated service policies, and inbound admission policies.

Authority requirements

Quality of service (QoS) policies might contain sensitive information about your network. Therefore, QoS administrative authority must only be granted when necessary.

The following authorities are required before you can configure QoS policies and (optionally) LDAP directory servers.

Grant authorities needed to manage the directory server

The QoS administrator needs the following authorities: `*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 window, 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 window.

System requirements

Quality of service (QoS) is an integrated part of the operating system.

You must complete these requirements:

1. Install TCP/IP Connectivity Utilities (5722-TC1).
2. Install iSeries Navigator on your PC. Make sure you install the Networking component during the iSeries Access installation. Quality of service is located under IP Policies within the Networking folder.

   Related concepts

   iSeries Navigator

   Related reference

   "Related information for QoS" on page 67
   Listed here are the IBM Redbooks (in PDF format), Web sites, and information center topics that relate to the quality of service (QoS) topic. You can view or print any of the PDFs.

Service-level agreement

This topic points out some of the important aspects of a service-level agreement (SLA) that might affect your quality of service (QoS) implementation.

QoS is a network solution. To receive network priority outside your private network, you might need to have an SLA with your ISP.

When is an SLA required

You only need a SLA if your policies require priority outside your private network. If you are using outbound policies to throttle traffic leaving your server, then no service guarantee is required. For example, on the server, you can create a policy that gives one application higher priority than another application. Your server recognizes this priority, but anything outside the server might not recognize the priority. If you have a private network and configure your routers to recognize codepoint markings (used to give outbound policies a service-level), then the routers will give priority through your private network. However, if the traffic leaves your private network, there are no guarantees. Without an SLA, you do not control how network hardware will handle traffic. Outside your private network, you will need a SLA to guarantee priority for a class of service or resource reservation.

Why is an SLA required

Your policies and reservations are only as good as the weakest link. This means, that QoS policies enable applications to receive priority through the network. However, if one node anywhere between the client and the server is unable to perform any of the traffic-handling characteristics discussed in the differentiated service or integrated service 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 QoS 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, what is dropped, marked, shaped, or retransmitted. The key reasons to provide QoS 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 might waste your resources. For example, if you ask to reserve 500 Kbps for IP telephony but your application only needs 20 Kbps, you might pay extra without receiving any notice from your ISP.

Note: QoS policies allow you to negotiate service-levels with your ISP which might decrease network service costs. For example, your ISP might be able to guarantee you a certain monetary rate if you do not exceed an agreed upon bandwidth level. Or you might state that using QoS policies, you will only use ″x″ amount of bandwidth during daytime hours, ″y″ amount of bandwidth at night, and agree to a rate for each time frame. Again, if the bandwidth is exceeded, the ISP might charge more. The ISP will need to agree to a certain service-level and have the ability to track the bandwidth you use.

Related concepts

"Concepts" on page 2
If you are new to quality of service (QoS), you can read some basic QoS concepts. This will give you an overview of how QoS works and how QoS functions work together.

"Scenario: Limit browser traffic" on page 28
You can use quality of service (QoS) to control traffic performance. Use a differentiated service policy to either limit or extend an application's performance within your network.

"Scenario: Secure and predictable results (VPN and QoS)" on page 32
If you are using a virtual private network (VPN), you can still create quality of service (QoS) policies. This example shows the two being used together.

"Scenario: Predictable B2B traffic" on page 38
If you need predictable delivery and still need to request a reservation, you also use an integrated service policy. This example uses a controlled load service.

"Scenario: Dedicated delivery (IP telephony)" on page 42
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, guaranteed service is used.

Network hardware and software

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

Applications

Integrated service policies require applications that are enabled by ReSerVation Protocol (RSVP). Because the iSeries applications are not presently RSVP-enabled, you must enable them to use RSVP. To enable your applications, you need to write special programs using the RSVP APIs or qtoq QoS sockets APIs. These programs will allow your applications to use RSVP.

Network nodes

The routers, switches, and even your own servers must have the capability to use QoS. To use differentiated services policies, your equipment must be enabled for differentiated services. This means that the network node must be able to classify, meter, mark, shape, and drop IP packets (traffic conditioners).

To use integrated services policies, your equipment must be RSVP-enabled. This means that the network nodes must also be able to support RSVP.

Related concepts

"QoS APIs" on page 16
You can read this topic to learn about protocols, APIs, and requirements for a router that is enabled for the ReSerVation Protocol (RSVP). The current quality of service (QoS) APIs include the RAPI API, the qtoq socket API, the sendmsg() API, and the monitor APIs.

"Traffic conditioners" on page 5
To use quality of service (QoS) policies, network equipment (like routers and switches) must have the capability for traffic conditioners. Traffic conditioners refer to classifiers, meters, markers, shapers, and droppers.

Configure QoS

You can use this topic to create differentiated service policies, integrated service policies, and inbound admission policies.

After you plan for quality of service (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 Uniform Resource Identifiers (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.

Enable QoS policies

Before your policies can take effect, they must be enabled. If you used the wizards, the server automatically enables the policies for you. However, if you changed a policy using the configuration objects, you need to dynamically update the server before the policies will become active. Before you enable them, be sure to look for overlapping policies that might cause problems.

Related concepts

"Plan for QoS" on page 48
The most important step to accomplishing quality of service (QoS) is planning. To receive expected results, you must review your network equipment and monitor network traffic.

iSeries Navigator

Related tasks

"Order QoS policies" on page 54
Whenever you have two policies that overlap, the physical order of your policies in iSeries Navigator is important.

Related reference

"Manage QoS" on page 54
You can use these procedures to manage existing quality of service (QoS) properties and policies.

Configure QoS with wizards

To configure quality of service (QoS) 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 a ReSerVation Protocol (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 carry out 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.

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 assign service-levels to outgoing traffic based on source/destination IP addresses, ports, applications, and even clients. In V5R3, your iSeries applications can receive levels of service based on more specific application information. For more information, see the differentiated service concept before creating this policy.

New Class of service wizard
Use the 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 differentiated service policy and an inbound admission policy.

New Inbound admission wizard
Use the Inbound admission wizard to restrict connections being made to your server. You can restrict access by TCP/IP address, by application, by local interface, or by Uniform Resource Identifier (URI). This allows a system administrator to control access to your server from specific clients, specific server applications, or by URI. In addition, you can enhance server performance.

Note: Before you set up an inbound policy that uses URIs, you must ensure that the application port assigned for the URI matches the listen directive enabled for Fast Response Cache Accelerator (FRCA) in the Apache Web Server configuration.

After you decide which type of policy to create, you can configure the policy by using the appropriate wizard previously listed.

Access the QoS wizards within iSeries Navigator

You can use these steps to access the QoS wizards and create a policy 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 click **Configuration**.

   Note: The Initial Configuration wizard opens in the following circumstances:
   - 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 steps in the Initial Configuration wizard. If the Initial Configuration wizard does not open, skip to step 4.
4. Select **Policies**. Right-click **IntServ**, **DiffServ**, or **Inbound admission**.
5. Select **New Policy**.

   Related concepts
   "QoS APIs" on page 16
   You can read this topic to learn about protocols, APIs, and requirements for a router that is enabled for the ReSerVation Protocol (RSVP). The current quality of service (QoS) APIs include the RAPI API, the qtoq socket API, the sendmsg() API, and the monitor APIs.

"Differentiated service" on page 2
This is the first type of outbound bandwidth policy you can create on your server. Differentiated service divides your traffic into classes. To carry out a differentiated service policy, you need to determine how you want to classify your network traffic and how to handle the different classes.

Related information

Manage addresses and ports for your HTTP server (powered by Apache).

Configure directory server

QoS policy configurations can be exported to an LDAP directory server.

This can make your quality of service (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 QoS 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 refer to the QoS policies
- Determine whether to use SSL security with your LDAP directory server
- Determine whether 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 i5/OS™ security auditing

If you are using iSeries Navigator, you will already have access to the default QoS Schema. The actual schema file is located on your server at `/QIBM/UserData/OS400/DirSrv`. However, if you are using an editor other than iSeries Navigator, you will need to import the LDIF file described in the following section. 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 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 pane. See LDAP concepts for a sample QoS schema.

Related concepts

"Directory server" on page 25
You can export your policies to a directory server. Read this topic to see the advantages of using or not using a directory server, Lightweight Directory Access Protocol (LDAP) concepts and configuration, as well as the quality of service (QoS) schema.

"Distinguished name" on page 26
When you want to manage part of your directory, you refer to the *distinguished name (DN)* or (if you choose) a keyword.

IBM Directory Server for iSeries (LDAP)

SSL security with your LDAP directory server

"Keywords" on page 25
When you configure your directory server, you need to determine whether to associate keywords with each quality of service (QoS) configuration.

Related information

iSeries LDAP Web page

Order QoS policies

Whenever you have two 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, schedule, local IP address, Uniform Resource Identifier (URI), server data, codepoint or protocol. 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 Policy Overlap window appears.

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

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

Related concepts

"Configure QoS" on page 51
You can use this topic to create differentiated service policies, integrated service policies, and inbound admission policies.

"Copy an existing policy" on page 55
Rather than create all of your policies from the beginning, you can make copies of the original policy and then edit the sections of the policy that differ from the original policy.

"Troubleshoot QoS" on page 61
Quality of service (QoS) provides several methods to troubleshoot QoS problems.

Related tasks

"Access QoS help in iSeries Navigator" on page 55
You can use iSeries Navigator to access the quality of service (QoS) help.

Manage QoS

You can use these procedures to manage existing quality of service (QoS) properties and policies.

These topics tell you where to find actual tasks for editing, enabling, viewing, and using other policy management techniques. There is also an explanation of how to use the QoS monitor and data collection to help analyze your IP traffic through the server.

Related concepts

"Configure QoS" on page 51
You can use this topic to create differentiated service policies, integrated service policies, and inbound admission policies.

Access QoS help in iSeries Navigator

You can use iSeries Navigator to access the quality of service (QoS) help.

To access the QoS help, you must use iSeries Navigator:

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

Related tasks

"Order QoS policies" on page 54
Whenever you have two policies that overlap, the physical order of your policies in iSeries Navigator is important.

Back up QoS policies

You should back up your quality of service (QoS) policies to eliminate the need to recreate your policies in the event of a server outage or power loss.

Your policies can be stored locally or exported to a directory server. You must specifically back up the following integrated file system directory: `QIBM/UserData/OS400/QOS/ETC, QIBM/UserData/OS400/QOS/TEMP, and QIBM/UserData/OS400/QOS/USR`. You must also back up 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 takes 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.**

   The *Backup and recovery* book provides instructions on conducting backups from integrated file systems.
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 online source to another.

Note: Your iSeries system copies information to the system disk, not to a diskette. 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 might 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.

Related information

Backup and Recovery

Copy an existing policy

Rather than create all of your policies from the beginning, you can make copies of the original policy and then edit the sections of the policy that differ from the original policy.

In iSeries Navigator, this quality of service (QoS) function is called *New based on*. You must use iSeries Navigator to access the QoS window 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 the policies, be sure to look for overlapping policies that might cause problems.

Related tasks

"Order QoS policies" on page 54
Whenever you have two policies that overlap, the physical order of your policies in iSeries Navigator is important.

Edit QoS policies

As your needs change, you must edit your policies to ensure that you are still receiving the appropriate performance.

You must attempt to correct any errors and make any necessary changes to your policies before activation. This is the best way to prevent complications with your policy results.

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 Uniform Resource Identifier (URI). These objects allow you to edit a policy.

To edit a policy in iSeries Navigator, follow the steps on the Editing a quality of service (QoS) policy page within the iSeries Navigator help.

Monitor QoS

You can use the quality of service (QoS) 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 monitor all active policies, select **Server** → **Monitor** from the QoS Configuration Server window. If you right-click a single policy and select **Monitor**, the monitor will only display information for that one policy.

You can use the monitor policies in the following ways:

- **To view real-time data on active policies**

  When you open the monitor, real-time data is always displayed on active policies. There is no need to start data collection.
- **To collect and save data over a period of time**

  If you want to save monitor results, then you need to start QoS data collection. The monitor continues to collect data until you stop the collection. Closing the monitor window does not stop the data collection. You can also change the properties that the monitor uses when collecting data. On the QoS Monitor window, highlight *QoS monitor* and select *File-->Properties* to change your options. Use the online help for additional information.

  If QoS data collection is turned on and monitor properties are changed, then you must perform the following steps to ensure that the changes are reflected in data collection.

  1. Stop QoS Data Collection.
  2. Change monitor properties.

    a. In the Monitor window, click **QoS Monitor**.
    b. Select **File** → **Properties**.
    c. Change the monitor properties and click **OK**.
3. Update the QoS Server.
4. Start 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), and Inbound admission. 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 given a definition: accepted requests, active connections, connections services, connection rates, dropped requests, in-profile packets, in-profile 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 about the monitor output field for each policy type. See any of the QoS scenarios for a sample of how to use the monitor along with the QoS policies.

Differentiated service policies

Table 4. 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 might 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.

Table 4. Differentiated service policies (continued)

Field	Description
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

IntServ policies do not display in the monitor until the applications are running and reservations have been established. If your IntServ policies have more than one reservation, you will see multiple entries in the monitor.

Table 5. 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 out-of-profile	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 in-profile	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 in-profile	The number of transmitted IP packets that fit within this policy's parameters.

Table 5. Integrated service (controlled load) policies (continued)

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

Integrated service (guaranteed) policies

IntServ policies do not display in the monitor until the applications are running and reservations have been established. If your IntServ policies have more than one reservation, you will see multiple entries in the monitor.

Table 6. 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 out-of-profile	The number of transmitted bits that exceed the policy's parameters.
Guaranteed rate	The guaranteed rate in bits per second.
Bits in-profile	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 in-profile	The number of transmitted IP packets that fit within this policy's parameters.
Slack term	The difference (in seconds) between the required delay and the delay obtained.
Source port range	The source port range which determines which applications are controlled by this policy.

Inbound admission policies

Table 7. Inbound admission 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.
Uniform Resource Identifier (URI)	The identity of the URI being policed.

Related concepts

"Scenario: Limit browser traffic" on page 28
You can use quality of service (QoS) to control traffic performance. Use a differentiated service policy to either limit or extend an application's performance within your network.

"Scenario: Secure and predictable results (VPN and QoS)" on page 32
If you are using a virtual private network (VPN), you can still create quality of service (QoS) policies. This example shows the two being used together.

"Scenario: Limit inbound connections" on page 36
If you need to control the inbound connection requests made to your server, use an inbound admission policy.

"Scenario: Predictable B2B traffic" on page 38
If you need predictable delivery and still need to request a reservation, you also use an integrated service policy. This example uses a controlled load service.

"Scenario: Dedicated delivery (IP telephony)" on page 42
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, guaranteed service is used.

"Scenarios" on page 28
These quality of service (QoS) policy scenarios can help you understand why and how to use QoS.

"Monitor server transactions" on page 63
With the quality of service (QoS) monitor, you can verify that the QoS policies are working as you intend them to work.

"Scenario: Monitor current network statistics" on page 46
Within the wizards, you are asked to set performance limits. These values cannot be recommended because they are based on individual network requirements.

Troubleshoot QoS

Quality of service (QoS) provides several methods to troubleshoot 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 might 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.

Enable QoS on the server

If the QoS server does not start, the first thing to check is whether QoS is enabled on the server. When you configure your policies for the first time, the Initial Configuration wizard automatically enables QoS on the server. However, if this value has been changed for any reason, the server will not start.

To verify that QoS is enabled on the server, follow these steps:

1. In iSeries Navigator, expand your server → **Network** → **IP Policies**.
2. Right-click **Quality of service** and select **Configuration**.
3. When the QoS interface appears, right-click **QoS** and select **Properties**.
4. On the QoS properties page, verify that **Enable QoS** is selected.

Related concepts

Communications trace

Related tasks

"Order QoS policies" on page 54
Whenever you have two policies that overlap, the physical order of your policies in iSeries Navigator is important.

Journal QoS policies

Quality of service (QoS) includes a journaling function. Journaling allows you to track QoS policy actions, such as when a policy was added, removed, or changed.

Journaling 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 a.m. to 4:00 p.m. You can check the journal log to see if the policy was actually added at 9:00 a.m. and removed at 4:00 p.m.

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, follow these steps:

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 box to turn journaling off.

Note: If the server is already started before you complete the steps above, you must stop and restart the server. After 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 way will reread the policy.conf file and look for the journaling attribute.

View the journal entries on the monitor

To view these journal entries, follow these steps:

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

View 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 quality of service (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 string 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 correct field descriptions.
   - DSPJRN JRN(QUSRSYS/QQOS) JRNCDE((M)) ENTTYP(MP) CMTCYCID(*ALL) OUTPUT(*OUTFILE) OUTFILFMT(*TYPE4) OUTFILE(*userlib/userfile*)
   - DSPF FILE(*userlib/userfile*)

Log QoS server jobs

When you encounter problems with your quality of service (QoS) policies, 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, follow these steps:

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

This opens a window that 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 the Work with Spooled File command (WRKSPLF QTCP) and look for the QTOQSRVR log.

Check the work spooled file for an error

To check the spooled file for an error, perform the following steps:

1. From a command-line interface, enter `WRKSPLF QTCP` and press Enter. The Work with All Spooled Files panel opens.
2. 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 Exit 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 panel, enter the following information and press Enter.

   ```
   Message File: QTCPMSG
   Library: *LIBL
   ```
7. On the Work with Message File panel, select **option 5** to display the message file you want to view and press Enter.
8. On the Display Message Descriptions panel, 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 required message ID and press Enter.
10. On the Select message details to display panel, select **option 30 (All of the Above)** and press Enter.

    A detailed description of the message appears.

Monitor server transactions

With the quality of service (QoS) monitor, you can verify that the QoS policies are working as you intend them to work.

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 might want to use the monitor before you even start any business-specific policies.

Try to create a differentiated service 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. If your policies are appearing in the monitor but do not appear to be affecting traffic, do the following verification:

- If the policy is filtering based on a Uniform Resource Identifier (URI), verify that Fast Response Cache Accelerator (FRCA) is enabled and configured properly. Before you set up an inbound policy that uses URIs, you must ensure that the application port assigned for the URI matches the listen directive enabled for FRCA in the Apache Web server configuration.
- Verify that the policy schedule. You might be looking for results during an inactive time.
- Verify that the port number is correct.
- Verify that the IP address is correct.

For some examples of monitor output, visit the QoS scenarios or view all the monitor fields in monitoring.

Related concepts

"Plan for QoS" on page 48
The most important step to accomplishing quality of service (QoS) is planning. To receive expected results, you must review your network equipment and monitor network traffic.

"Scenarios" on page 28
These quality of service (QoS) policy scenarios can help you understand why and how to use QoS.

Related reference

"Monitor QoS" on page 56
You can use the quality of service (QoS) monitor to analyze your IP traffic through the server.

Related information

Manage addresses and ports for your HTTP server (powered by Apache)

Trace TCP applications

You can use the quality of service (QoS) trace to work with trace functions and to view the current trace buffer.

To run the trace on the server, type TRCTCPAPP (Trace TCP/IP Application command) from a command-line interface.

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 it 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 (MAXSTG)	1-16000, *APP
Trace full action (TRCFULL)	*WRAP, *STOPTRC

Settings	Options
Argument lists (ARGLIST)	Levels: ’lvl=1’, ’lvl=2’, ’lvl=3’, ’lvl=4’ Content: ’c=a’, ’c=i’, ’c=d’ , ’c=m’
QoS trace type	*ALL

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=4 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. In a typical trace, it is recommended you specify ’l=4’.

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 might occur if you are running out of system memory or if your system cannot communicate with TCP/IP. This is the default level.

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

Level 3: Specific Events (EVENT)
Logs any QoS operation that has occurred. For example, an event log records 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 might use this high-level trace for logging anything that you think might be helpful for debugging problems. This information is helpful to determine where a problem occurred and how to reproduce the problem.

Trace content

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.

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.

If you need help interpreting the trace output, read the trace output example on the trace output page, which contains sample output with comments to help you interpret its meaning. The TRCTCPAPP function is typically used by the service, so if you have problems reading the output, you might contact your service representative.

Related reference

TRCTCPAPP (Trace TCP/IP Application) Command Description

Examples: 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 pay attention to is whether the ReSerVation Protocol (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 can 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

Listed here are the IBM Redbooks (in PDF format), Web sites, and information center topics that relate to the quality of service (QoS) topic. You can view or print any of the PDFs.

QoS Request for Comments (RFCs)

Requests for Comments (RFCs) are written definitions of protocol standards and proposed standards used for the Internet. The following RFCs might be helpful for understanding QoS and its related functions:

- **RFC 1349.**

  This RFC discusses the new definition of the type of service octet field in an IP packet header.
- **RFC 2205.**

  This RFC explains the definition of ReSerVation Protocol (RSVP).
- **RFC 2210.**

  This RFC explains the use of RSVP with IETF Integrated Services.
- **RFC 2474.**

  This RFC explains the definition of the Differentiated Services Field (DS Field).
- **RFC 2475.**

  This RFC explains the architecture of differentiated services.

To view the RFCs previously listed, visit the RFC Index Search Engine located on the RFC Editor Web site.

IBM Redbooks

- iSeries IP Networks: Dynamic! (about 16 589 KB). This is the most recent IP networking redbook. It shows you how to design an IP network that is self-configuring, fault tolerant, and efficient in its operation. In addition to many other functions, it explains both the theory behind QoS and its implementation on the iSeries. You will also find more scenarios with step-by-step instruction.

- TCP/IP More Cool Things than Ever (about 10 035 KB). 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 include QoS yet, but it does go through LDAP directory server information.

- TCP/IP Tutorial and Technical Overview (about 7885 KB). 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 QoS within *Part 3. Advanced concepts and new technologies* under Chapter 22.

Other information

- Directory services (LDAP). This topic covers directory server basics, configuration, administration, and troubleshooting. The directory services topic also provides additional resources for configuring your directory server.

- Intrusion detection. This topic discusses gathering information about unauthorized access attempts and attacks coming in over the TCP/IP network. Security administrators can analyze the auditing records that intrusion detection provides to secure the iSeries network from these types of attacks.

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

Software Interoperability Coordinator, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, IBM License Agreement for Machine Code, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming Interface Information

This Quality of Service publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM i5/OS.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM
IBM (logo)
iSeries
i5/OS
Redbooks

Other company, product, and service names may be trademarks or service marks of others.

Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

Personal Use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these publications, or any portion thereof, without the express consent of IBM.

Commercial Use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED ″AS-IS″ AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Printed in USA