Keywords: HQoS, multi-level scheduling, WFQ, WRED, GTS

Abstract: Hierarchical QoS (HQoS) is a technology guaranteeing bandwidth for multiple services and multiple users in Diff-Serv model through hierarchical scheduling. This document introduces the evolution, key technical issues, and typical applications of HQoS.

Acronyms:

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full spelling</th>
</tr>
</thead>
<tbody>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>H-QoS</td>
<td>Hierarchical QoS</td>
</tr>
<tr>
<td>SLA</td>
<td>Service Level Agreement</td>
</tr>
<tr>
<td>WFQ</td>
<td>Weighted Fair Queue</td>
</tr>
<tr>
<td>CBWFQ</td>
<td>Class-Based Weighted Fair Queue</td>
</tr>
<tr>
<td>ACL</td>
<td>Access Control List</td>
</tr>
<tr>
<td>WRED</td>
<td>Weighted Random Early Drop</td>
</tr>
<tr>
<td>TM</td>
<td>Traffic Management</td>
</tr>
<tr>
<td>PHB</td>
<td>Per Hop Behavior</td>
</tr>
<tr>
<td>DSCP</td>
<td>Differentiated Service Code Point</td>
</tr>
<tr>
<td>NP</td>
<td>Network Processor</td>
</tr>
<tr>
<td>Diff-Serv</td>
<td>Differentiated Service</td>
</tr>
<tr>
<td>Inter-Serv</td>
<td>Integrated Services</td>
</tr>
<tr>
<td>MQC</td>
<td>Modular QoS Command line</td>
</tr>
</tbody>
</table>
# Table of Contents

1 Overview ......................................................................................................................................... 3  
  1.1 Background .......................................................................................................................... 3  
  1.2 Benefits .................................................................................................................................. 4  
    1.2.1 Finer Per-User, Per-Service QoS Granularity ........................................................... 4  
    1.2.2 Reflecting Network Topology and Traffic Forwarding Paths ............................... 5  
    1.2.3 Configuring QoS on a Single Point ........................................................................... 5  
    1.2.4 Mature Traffic Accounting Function and Reasonable Per-Service Bandwidth Allocation ............................................................ 5  
  1.3 HQoS Model ........................................................................................................................ 6  

2 HQoS Implementation on SR8800 ................................................................................................. 7  
  2.1 Dedicated QoS Engine ........................................................................................................ 7  
  2.2 Hierarchical Scheduling Model ............................................................................................ 8  

3 Technical Characteristics .............................................................................................................. 10  
  3.1 Schedulers and Queues .................................................................................................... 10  
  3.2 Traffic Classification and Queue Mapping ......................................................................... 10  
  3.3 Hierarchical Queue Scheduling ......................................................................................... 11  
  3.4 Traffic Shaping ................................................................................................................... 11  
  3.5 Congestion Avoidance ....................................................................................................... 12  
  3.6 Traffic Accounting, Diagnosis, and Maintenance ............................................................ 12  
  3.7 Support for Dynamic QoS Parameters Application and Tuning ......................................... 12  

4 Application Analysis ...................................................................................................................... 13  

5 Application Scenarios ................................................................................................................... 14  
  5.1 Network Requirements ...................................................................................................... 14  
  5.2 Configuration Considerations ......................................................................................... 16  
  5.3 Configuration Procedure .................................................................................................. 16  

6 Summary ...................................................................................................................................... 18  

7 References ................................................................................................................................... 18
1 Overview

1.1 Background

Along with the development of the Internet, more and more services are emerging on the Internet. The requirements of these services are diversified. For example, real-time applications, Voice over IP (VoIP) for example, require low transmission delay while E-mail and FTP applications are insensitive to transmission delay. To satisfy diversified requirements of different service applications, a network device must be able to identify different applications and then provide differentiated network services for them. QoS was thus introduced. The evolution of QoS is shown below:

- **Best-effort**: In the 1980s, the Internet mainly carried data traffic. At that time, the Internet provided best-effort services without any QoS guarantee.

- **Flow-based Inter-Serv QoS**: Driven by real-time applications such as VoIP, the IETF launched an RSVP-based Inter-Serv solution in 1994. Unfortunately, this end-to-end flow-based QoS technology was not widely used in the IP field.

- **Class-based Diff-Serv QoS**: To implement scalability and simplicity, the IETF launched the Diff-Serv model, a class-based QoS technology, in 1998. The Diff-Serv model defines a finite set of classes for traffic. At the boundaries of a Diff-Serv network, a packet is assigned to a service class according to its service requirements, and then regulated based on the policy associated with the class and assigned a DSCP precedence. In the network, all nodes handle the packet in compliance with the PHB determined by the DSCP. IP Diff-Serv is a widely used QoS technology now.

However, as the expansion of single-port capacity due to the development of networking devices makes a port provide access for more and more users, traditional QoS encounters the following problems:

1. Traditional traffic management schedules traffic based on port bandwidth, which is sensitive to service levels but not to users. Therefore, it can work well at the network side but not at the service access side.

2. It is hard for traditional traffic management to manage traffic on a per-service and per-user basis.

To address the problems and provide finer-granularity QoS, Hierarchical QoS (HQoS)
was developed. HQoS adopts hierarchical scheduling and can not only manage traffic on a per-user basis but also schedule user traffic based on priority. With HQoS and the optimized hardware architecture, the SR8800 can perform internal resources control policies to provide guaranteed QoS services for advanced users and save the overall networking cost.

1.2 Benefits

1.2.1 Finer Per-User, Per-Service QoS Granularity

The generic QoS performs port-level queue scheduling, which is sensitive to service levels but not to users. As a result, the preemption results may be undesirable. As shown in Figure 1, HQoS adopts hierarchical scheduling to address this problem.

![Figure 1](image)

**Figure 1** Queue scheduling model based on a per-user, per-service basis

The above figure illustrates how HQoS performs bandwidth management for multiple services of multiple users. As shown in the figure, an interface is like a big pipe, and each user gets a small pipe from the big pipe. In this way, the big pipe can perform traffic management for the contained small pipes, and each user can subdivide its small pipe into channels for different application flows and manage them differently.
The idea is implemented through hierarchical scheduling: Level-1 scheduling is performed based on users, and level-2 scheduling is performed based on the services of a user.

1.2.2 Reflecting Network Topology and Traffic Forwarding Paths

A typical IP network consists of the access layer, the distribution layer, and the backbone layer. HQoS fully reflects this hierarchy, where the access layer connects users to the network through various access methods, the distribution layer distributes the traffic aggregated from the access layer, and the backbone layer fast forwards the traffic. The network topology has the following features:

1. The topology is usually tree shaped. Even for some networks that are rings at the physical layer, tree topology is adopted at the link layer.
2. The forwarding path of each service tends to be fixed and is not prone to dynamic changes.
3. A service router is used as the root node in the tree topology.

1.2.3 Configuring QoS on a Single Point

Usually, the LAN switches attached to a service router forward packets only based on Layer 2 information and thus their QoS capabilities are inadequate for diversified services. You can surely address the issue by replacing the Layer-2 switches on the access network with higher-class devices, but the investment can be huge and the effect is not necessarily desirable. On the contrary, HQoS is a cost-effective solution. HQoS can reflect network topology and the forwarding path of each service. You can configure QoS on the traffic convergence point (that is, the service router) and then map the networking devices and service types attached to the router to nodes at a certain layer on the convergence point. Thus, QoS configured on a node reflects the QoS capability of the associated device.

1.2.4 Mature Traffic Accounting Function and Reasonable Per-Service Bandwidth Allocation

HQoS provides a mature traffic accounting function, which enables you to manage traffic better. With the traffic accounting function, you can view the bandwidth allocated to each service, and thus allocate bandwidth for each service appropriately based on traffic analysis.
1.3 HQoS Model

The DSL Forum suggests a generic HQoS model (as shown in Figure 2) in its TR-059 *DSL Evolution – Architecture Requirements for the Support of QoS-Enabled IP Services*. In TR-101 (Technical Report DSL Forum TR-101 Migration to Ethernet-Based DSL Aggregation), the DSL Forum clarifies the application of the model in Ethernet.

![Generic HQoS model suggested by the DSL Forum](image)

**Figure 2** Generic HQoS model suggested by the DSL Forum

In this model, the edge devices provide multiple levels of congestion management capabilities to avoid congestion on the downstream devices incapable of QoS and guarantee bandwidth for high-priority traffic. Congestion management is implemented through multiple levels of queues and schedulers with the proper queue scheduling algorithms. Simply speaking, each scheduler provides several packet queues, where the packets from the upper level scheduler wait for scheduling and forwarding.

Through configuring proper queue and scheduler parameters based on the bandwidth requirements and traffic priority, you can ensure that traffic on the edge devices can be scheduled properly and high-priority traffic can be processed preferentially.
2 HQoS Implementation on SR8800

2.1 Dedicated QoS Engine

As increasingly more network services are emerging, routers have to consume lots of resources and spend more time to handle increased QoS requests. This has been one of the major factors that hinder the improvement of router performance. Using a single chip to process traffic and QoS simultaneously has been far from satisfying the requirements.

Considering that QoS scheduling requirements are stable and the QoS model is mature, you can improve QoS scheduling performance by frozen the function into the ASIC. The same is true of the table lookup function.

Integrating high performance of ASICs and flexible service expansion of NPs, the SR8800 router adopts the triple-engine forwarding architecture, which consists of the NP service engine, the QoS engine, and the table lookup engine. For the service engine, NPs are adopted to achieve flexible service expansion and upgrade. For the QoS engine and the table lookup engine, ASICs are adopted to achieve high performance in QoS processing and table lookup.
2.2 Hierarchical Scheduling Model

The hierarchical scheduling model consists of the port layer, the user group layer, the user layer, and the user service layer. The HQoS processing engine can perform congestion avoidance, multi-level scheduling, rate limiting, and traffic accounting.

![Hierarchical scheduling model](image)

**Figure 4** Hierarchical scheduling model

(1) User service scheduler

The user service scheduler is used to process various types of service traffic of a user. The SR8800 router supports up to four service types for a user. The traffic of each user is classified into the specified service types and then assigned to different hardware queues. The service scheduler dequeues packets from these queues and then sends the packets to the upper level scheduler.

The service scheduler adopts the Strict Priority (SP) algorithm and the Weighted Fair Queuing (WFQ) algorithm. The SP algorithm schedules packets in the descending order...
of queue priority. Packets in a queue are scheduled until all higher-priority queues are empty. WFQ schedules queues with the same priority according to their weights. Usually, SP is used to guarantee the forwarding of high-priority traffic, while WFQ is used to allocate per-flow bandwidth in the ratio of weights.

The scheduler keeps monitoring the queue buffer usage status, and drops packets actively when congestion tends to deteriorate. In this way, the scheduler adjusts the network traffic so as to solve the overloading problem in the network. On this layer, WRED parameters, including the upper threshold and lower threshold are configured for each queue based on drop precedence. WRED can randomly drop exceeding packets and thus avoid congestion.

You can configure WFQ and WRED parameters for the user service scheduler.

(2) User scheduler

The user scheduler performs traffic shaping for all traffic of a user, dequeues these packets, and sends them to the upper level scheduler after processing.

Traffic shaping uses token buckets for traffic specification evaluation. The packets exceeding the specification are buffered so that they are not dropped, and thus the traffic is smoothed. In this way, a user performs service level agreement (SLA) management for the traffic. The user scheduler supports shaping algorithms RFC SrTCM and TrTCM, and parameters such as CIR, PIR, CBS, and EBS.

The CIR supported by the user scheduler ranges from 128 kbps to 10 Gbps, at the step of 1 kbps.

(3) User group scheduler

The user group scheduler is used to process traffic of a user group or logical subinterface. The packets sent to user groups or different subinterfaces are assigned to different queues. The user group scheduler dequeues packets from these queues, schedules them, and then sends them to the upper level scheduler.

The user group scheduler uses WFQ or class-based WFQ (CBWFQ) for queue scheduling and can assign different weights to different user groups. The user group scheduler supports traffic shaping, and the CIR supported by the user group scheduler ranges from 128 kbps to 10 Gbps, at the step of 1 kbps.
(4) Port-level scheduler

The port-level scheduler is used to process traffic of physical interfaces on the interface boards. It supports traffic shaping and traffic accounting.

The packets sent to different physical interfaces are assigned to different port-level queues. The port-level scheduler dequeues packets from these queues and sends them out the corresponding physical interfaces. The port-level scheduler uses WFQ or CBWFQ for queue scheduling. If an interface card provides 10×GE interfaces, the port-level scheduler schedules the ten port-level queues in turn. The port-level scheduler supports traffic shaping, and the CIR supported by the port-level scheduler ranges from 10 Mbps to 10 Gbps, at the step of 10 Mbps.

3 Technical Characteristics

3.1 Schedulers and Queues

The hierarchy of schedulers determines how adaptive HQoS to complex topologies: the more scheduler levels, the more service types and users, and thus the more application scenarios. A typical HQoS hierarchy consists of four levels of schedulers, that is, user service schedulers, user schedulers, user group schedulers, and port-level schedulers.

Queues are units for implementing various QoS functions, such as rate limiting, congestion avoidance, traffic shaping, and queue scheduling. The number of queues determines the maximum number of flows that HQoS can process simultaneously, and the depth of a queue determines the maximum number of packets that can be cached in the queue. Generally, HQoS is required to cache packets transmitted at wire speed in 200 ms. An SR8800 router provides 16-k queues per NP with 512 MB cache, which are adequate for accommodating bursty traffic and performing traffic shaping.

3.2 Traffic Classification and Queue Mapping

To perform high-priority services for high-priority traffic and satisfy the requirements of real-time traffic for delay and jitter, a router must be capable of identifying service types. The SR8800 supports classifying traffic based on 802.1p priority, DSCP, EXP, IP quintuple, and so on.

In the SR8800, NP processing is separated from QoS scheduling to improve service
processing and QoS scheduling performance and increase the number of identifiable service types. The SR8800 implements traffic class-to-queue mapping through global mapping table configuration and thus implements fast QoS deployment.

### 3.3 Hierarchical Queue Scheduling

In HQoS, different priorities or weights are assigned for schedulers and queues to ensure that when congestion is occurring, high-priority traffic can still be forwarded in approximate real-time and queues of the same level can be scheduled fairly. Usually, four user service queues are assigned to one group, which corresponds to a scheduler. The four queues are configured with different priorities and scheduled by PQ, LLQ, WFQ, or CBWFQ, so that high-priority traffic can be forwarded preferentially.

### 3.4 Traffic Shaping

The traffic a router sends out may exceed the capability of the links of the access network, or the low-priority queues may be starved to death when there is too much high-priority traffic. To address these problems, you should perform traffic shaping on each layer. Traffic shaping uses token buckets for traffic specification evaluation. The packets exceeding the specification are buffered so that they are not dropped, and thus the traffic is smoothed.

In the HQoS model, you can configure traffic shaping on multiple layers to implement granular bandwidth management and guarantee SLA for users and their traffic.

![Traffic shaping in the HQoS model](image.png)

**Figure 5** Traffic shaping in the HQoS model
3.5 Congestion Avoidance

As a type of flow control mechanism, congestion avoidance monitors the utilization status of the queue buffer, and can actively drop packets when congestion tends to deteriorate. In this way, the congestion avoidance mechanism adjusts the network traffic to prevent the network from being overloaded. Traditionally, tail drop is adopted. In this approach, when the queue length reaches the upper threshold, all the newly arriving packets are dropped. Tail drop may cause global TCP synchronization. To address this problem, you can use the Weighted Random Early Detection (WRED) mechanism. WRED sets an upper threshold and lower threshold for each queue, and processes the packets in a queue as follows:

- When the queue size is shorter than the lower threshold, no packet is dropped.
- When the queue size reaches the upper threshold, all subsequent packets are dropped.
- When the queue size is between the lower threshold and the upper threshold, the received packets are dropped randomly. The longer a queue is, the higher the drop probability is. However, a maximum drop probability exists.

In HQoS, to satisfy the traffic control requirements, you just need to perform congestion avoidance by configuring proper drop policies on the user service layers corresponding to the physical queues.

3.6 Traffic Accounting, Diagnosis, and Maintenance

During HQoS processing, the SR8800 can count forwarded traffic and dropped traffic both in packets and in bytes for the specific layer. At the same time, the status information of queue buffer depths, schedulers, and shapers is available for you to judge the running status of the hierarchical system. The SR8800 supports traffic accounting and displaying queue depths hierarchically.

3.7 Support for Dynamic QoS Parameters Application and Tuning

The QoS requirements of users and services on a network may change when the applications or network topology change. HQoS is required to adapt to such changes.
The SR8800 allows you to dynamically apply and modify the QoS parameters including WRED, PQ, LLQ, WFQ, CBWFQ, traffic shaping, and traffic accounting.

4 Application Analysis

Before implementing HQoS, you should make an overall QoS design following these guidelines:
• Analyze the services types to be implemented. In an enterprise network, the service types that should be considered include at least production data, video monitoring traffic, IP telephone traffic, and OA traffic. In a MAN, the service types that should be considered include VoIP traffic, video traffic, dedicated line traffic, and Internet traffic.

• Analyze the network topology, breaks the network down into multiple levels properly, identify the appropriate HQoS deployment node, and map the hierarchical nodes in the HQoS model to the devices in the actual network.

• Configure QoS policies appropriate to the users and service types to implement SLA for them successfully.

5 Application Scenarios

5.1 Network Requirements
Figure 6 Network diagram for HQoS configuration
### Service requirements and design

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>The service types of each user are basically the same, and different users need different traffic management policies.</td>
<td></td>
</tr>
<tr>
<td>- Limit the outbound rate of the PE router (the SR8800) to 155 Mbps.</td>
<td></td>
</tr>
<tr>
<td>- Set CIR to 75 Mbps and PIR to 100 Mbps for all traffic of user group 1.</td>
<td></td>
</tr>
<tr>
<td>- The production data of user 1 is on the network segment 10.1.1.0/24 and with priority 5.</td>
<td></td>
</tr>
<tr>
<td>- The telephone traffic of user 1 is on the network segment 10.1.2.0/24 and with priority 4.</td>
<td></td>
</tr>
<tr>
<td>- The monitoring traffic of user 1 is on the network segment 10.1.3.0/24 and with priority 3.</td>
<td></td>
</tr>
<tr>
<td>- The OA traffic of user 1 is on the network segment 10.1.4.0/24 and with priority 0.</td>
<td></td>
</tr>
<tr>
<td>- The network segments to which the other users belong are omitted here.</td>
<td></td>
</tr>
</tbody>
</table>

### 5.2 Configuration Considerations

- Perform differentiated queue scheduling policies for traffic of different users by assigning different users to different forwarding groups.
- Because the user groups to be used are the same in configuration structure, assign them to their corresponding forwarding groups through instantiation.
- Implement forwarding class mapping for user groups and user services through traffic classification and remarking. Mark different users with different QoS-local-IDs, and associate a service type with a forwarding class to assign different types of packets to different queues.

### 5.3 Configuration Procedure

1. Configure forwarding profiles for user groups. Take user group 1 for example here. Set the CIR to 75 Mbps and PIR to 100 Mbps.

```
[SR8800] qos forwarding-profile UserGroup_1
[sr8800-fp=UserGroup_1] gts cir 75000 pir 100000
```
(2) Configure forwarding profiles for users. Take user 1 in user group 1 for example here. Set the CIR to 15 Mbps and PIR to 20 Mbps.

[SR8800] qos forwarding-profile UserGroup_1_user1
[SR8800-fp-UserGroup_1_user1] gts cir 15000 pir 20000

(3) Nest Layer-2 forwarding groups in the Layer-1 forwarding group and specify a forwarding profile for each nested forwarding group. Take user 1 and user n in user group 1 for example.

[SR8800] qos forwarding-group UserGroup_1
[SR8800-fg-UserGroup_1] forwarding-group UserGroup_1_user1 profile UserGroup_1_user1
[SR8800-fg-UserGroup_1] forwarding-group UserGroup_1_usern profile UserGroup_1_usern

(4) Create a scheduling policy HQoS_Demo and instantiate it.

First create a user-defined scheduling policy named HQoS_Demo.

[SR8800] qos scheduler-policy HQoS_Demo

Associate a forwarding group with a forwarding profile in the scheduling policy. Take user group 1 for example here.

[SR8800-sp-HQoS_Demo] qos forwarding-group UserGroup_1 profile UserGroup_1

Instantiate the Layer-1 forwarding groups.

[SR8800-sp-HQoS_Demo] layer 1
[SR8800-sp-HQoS_Demo-layer1] qos forwarding-group UserGroup_1 group
[SR8800-sp-HQoS_Demo-layer1] qos forwarding-group UserGroup_n group

Instantiate the Layer-2 forwarding groups.

[SR8800-sp-HQoS_Demo-layer1] layer 2
[SR8800-sp-HQoS_Demo-layer2] qos forwarding-group UserGroup_1_user1 match qos-local-id 1
[SR8800-sp-HQoS_Demo-layer2] qos forwarding-group UserGroup_1_user2 match extended qos-local-id 2 to 99
[SR8800-sp-HQoS_Demo-layer2] qos forwarding-group UserGroup_n_user1 match extended qos-local-id 100 to 199

(5) Apply the scheduling policy to interface G 1/1/1.

[SR8800-G1/1/1] qos apply scheduler-policy HQoS_Demo outbound

(6) Configure an MQC policy named MQC_HQoS_Demo to classify the traffic into multiple classes based on user and service and mark each class with the corresponding QoS-local-ID and FC.

Omitted
(7) Apply the MQC policy to the outbound direction of interface G 1/1/1.

```
[SR8800-G1/1/1] qos apply policy MQC_HQoS_Demo outbound
```

6 Summary

Consolidation of telephone, monitoring, OA, and production networks is today's development trend. Data communications devices are hence required to carry more and more real-time delay-sensitive traffic such as voice and video. HQoS effectively address the requirements of data communications devices for providing differentiated QoS services for different services of different users. It has been an essential technology for building an elaborate data network.

HQoS implemented on a H3C SR8800 10G core router can be expanded to support thousands of services, with each slot providing up to tens of thousands queues. Services can be dynamically assigned to these queues. You can move a scheduler level up or down in the HQoS hierarchy dynamically as needed. An upper level scheduler controls the total bandwidth of the group of lower level schedulers. An upper level scheduler configures CIR and PIR for each lower level scheduler according to their levels and weights, thus implementing QoS for multiple services.

7 References

- TR-101, *Migration to Ethernet-Based DSL Aggregation*, DSL Forum