



The Microsoft QoS Components

White Paper

Abstract

In Part I of this white paper, *A Short Overview of QoS Mechanisms and Their InterOperation*, we described emerging QoS mechanisms and their application. We showed that the optimal QoS-enabled network relies on the cooperation of the host and the network. Microsoft is committed to enabling broad deployment of QoS-enabled networks by providing an extensive suite of QoS components. These include QoS-aware applications, QoS functionality in host operating systems, and network-based QoS components. Microsoft's QoS components are the topic of this part of the white paper.

© 1999 Microsoft Corporation. All rights reserved.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

Microsoft, Active Directory, MSN, NetMeeting, Windows, Windows Media, and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Other product and company names mentioned herein may be the trademarks of their respective owners.

Microsoft Corporation • One Microsoft Way • Redmond, WA 98052-6399 • USA

11/99

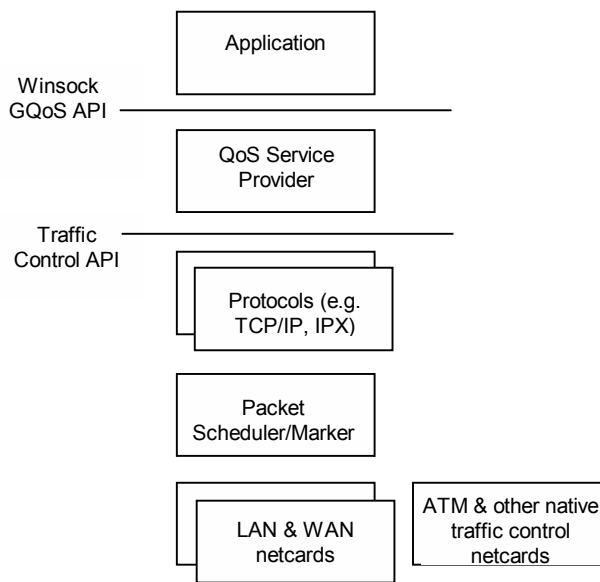
1	THE MICROSOFT QOS COMPONENTS.....	1
2	THE HOST PROTOCOL STACK	1
2.1	APPLICATION	1
2.2	WINSOCK2 & GQoS API.....	1
2.3	THE QoS SERVICE PROVIDER	2
2.3.1	<i>RSVP Signaling</i>	2
2.3.2	<i>QoS Policy Support</i>	2
2.3.3	<i>Invocation of Traffic Control</i>	2
2.4	THE TRAFFIC CONTROL API.....	3
3	DIFFSERV (DSCP) AND 802.1P MARKING BEHAVIOR.....	3
4	THE SUBNET BANDWIDTH MANAGER AND ADMISSION CONTROL SERVICE.....	5
4.1	THE SUBNET BANDWIDTH MANAGER.....	5
4.2	THE LOCAL POLICY MODULE AND EXTENSIBILITY	5
4.3	VARIATIONS OF THE ACS	6
5	THE ACTIVE DIRECTORY QOS SCHEMA.....	6
6	SUMMARY.....	7
6.1	FOR MORE INFORMATION.....	7

1 The Microsoft QoS Components

In Part I of this white paper, *A Short Overview of QoS Mechanisms and Their InterOperation*, we described emerging QoS mechanisms and their application. We showed that the optimal QoS-enabled network relies on the cooperation of the host and the network. Microsoft is committed to enabling broad deployment of QoS-enabled networks by providing an extensive suite of QoS components. These include QoS-aware applications, QoS functionality in host operating systems, and network-based QoS components. Microsoft's QoS components are the topic of this part of the white paper.

2 The Host Protocol Stack

The following diagram illustrates the host protocol stack of the Windows® 2000 operating system (all functionality above the line marked *Traffic Control API* is also available in the Windows 98 operating system):



In the following paragraphs, we'll describe each QoS related component.

2.1 Application

Microsoft's QoS-enabled applications include NetMeeting® conferencing software (included with Windows 2000 and Windows 98), Windows Media™ Services (to be released post Windows 2000) and TAPI –3.0 (though not an application in itself, TAPI –3.0 provides QoS services to numerous third party telephony applications). In addition, Microsoft has worked with the Enterprise Resource Planning (ERP) application vendor, SAP, to QoS-enable the SAP-R/3 kernel. The multimedia applications described fall into the category of *quantitative* applications (having readily quantifiable resource requirements), while SAP-R/3 is representative of *qualitative* applications (for which resource requirements are not readily quantifiable). By focusing on this short list early in the Windows 2000 release cycle, Microsoft is able to validate the full breadth of its QoS functionality. With the imminent release of Windows 2000, Microsoft will step up efforts to work with a broad range of ISVs to QoS-enable additional applications.

2.2 Winsock2 & GQoS API

A subset of the Winsock2 API is the *Generic QoS API* (GQoS). The purpose of this API is to enable applications to easily invoke QoS functionality. The API abstracts the complexity of the various

underlying QoS mechanisms and different network media, requiring only very simple directives from the application. For applications that are QoS savvy, extensions to the API provide more direct control of the underlying QoS mechanisms. In addition to providing the means by which applications invoke QoS functionality, the GQoS API provides feedback to the application regarding the status of the network.

Microsoft recommends that applications be ported to the GQoS API whenever the application is both:

- Generally deemed important to manage by network administrators, because it is mission-critical in nature, requires quality guarantees from the network and/or stands to have an adverse impact on network resources.
- Persistent in the sense that it is session-oriented and that the flows it establishes persist for a sufficient duration to justify the overhead of signaling.

A minimal port of an application (that already uses the Winsock2 API) requires about one day of programming work. More sophisticated use of the API, for example, integration of the application's UI with the QoS mechanisms and the ability to respond to network events, requires additional work.

The GQoS API invites application programmers to adopt a new perspective of the network. Traditionally, many network applications (especially multimedia) attempt to seize as many resources as possible from the network. Often, the consequences of this approach are that the application's deployment is prohibited or restricted by network administrators. By using the GQoS API, applications are able to request the resources they need and are either assured of these resources or given the equivalent of a "busy signal", depending on current resource availability and network administrator's policies. In response to a "busy signal", applications may be allowed to send but with no assurance of a quality guarantee, or alternatively, may be required to refrain from sending. As such, applications writers are encouraged to cooperate with network policies rather than to attempt to subvert them.

2.3 The QoS Service Provider

The QoS service provider (QoS SP) is the entity that responds to the GQoS API. It provides the following services:

- RSVP signaling
- QoS policy support
- Invocation of traffic control

2.3.1 RSVP Signaling

RSVP signaling is generated by default on behalf of applications using the GQoS API. The QoS SP initiates and terminates all RSVP signaling on behalf of the applications. It provides status regarding network QoS to applications that are interested, but does not require the application to understand RSVP signaling. As part of its RSVP signaling functionality, the QoS SP behaves as a Subnet Bandwidth Manager (SBM) client (SBM functionality will be explained in further detail later in this white paper).

2.3.2 QoS Policy Support

The QoS SP inserts a Kerberos authenticated Windows NT® user ID into RSVP signaling messages. In addition, the QoS SP inserts any application identification provided by the application via the GQoS API. The inserted objects identify the user and application such that policy decision points (PDPs) and policy enforcement points (PEPs) in the network can apply policy based on these objects.

2.3.3 Invocation of Traffic Control

The QoS SP actually enforces network policy by invoking traffic control in accordance with the network's response to RSVP signaling. In general, the QoS SP identifies two types of traffic control: greedy and non-greedy. Non-greedy traffic control is invoked in immediate response to an application's request for QoS.

Greedy traffic control is enabled only if (and to the degree) approved by the network, in response to RSVP signaling. These types of traffic control will subsequently be defined in further detail.

2.4 The Traffic Control API

The traffic control (TC) API offers the QoS SP and third party traffic management applications an interface by which to invoke kernel TC functionality. TC functionality includes the ability to create *flows* (which affect transmitted traffic) and to define classification criteria (which determine the set of packets directed to each flow). A flow has certain characteristics that are applied to all packets on the flow. These include packet scheduling or queuing behavior (such as transmission rate) and packet marking (such as 802.1p and DSCP marking), which determine the priority allotted to the packets in certain parts of the network.

The TC API is used by the QoS SP on behalf of QoS-aware applications. It may also be used by network management applications designed to directly invoke traffic control on hosts. To this end, the TC API is scriptable and remoteable. In invoking the TC API directly, traffic management applications bypass the policy control offered by the QoS SP. Therefore, these applications should be integrated with the same policy decision points in the network that effect policy via the QoS SP.

Note that, at present, the TC API and the corresponding functionality is applicable to transmitted traffic only. In future versions of Windows operating systems, TC functionality will be available to control the treatment of received traffic as well as transmitted traffic. Traffic control functionality is available in Windows 2000, but not in Windows 98 (with the exception of limited DSCP marking).

The TC API separates traffic control *consumers* from traffic control *providers*. Traffic control providers include kernel mode modules that implement any traffic control functionality in response to the traffic control API. Native traffic control functionality available in Windows 2000 includes:

- Packet scheduling.
- 802.1p marking for prioritization on LANs.
- DSCP marking for prioritization in routed networks.
- ISSLOW link layer fragmentation (per PPP multilink) for latency reduction on slow links.
- ATM VC control and cell scheduling.

The packet scheduler component provides traffic control functionality over any media that presents a LAN interface to the network stack, including Ethernet, token ring, FDDI and ATM LAN emulation. It is responsible for packet scheduling as well as 802.1p and DSCP marking. The packet scheduler also provides ISSLOW functionality over slow WAN adapters. Microsoft's classical IP over ATM (CLIP) provides traffic control functionality without requiring the packet scheduler. Additional traffic control providers planned for the future include cable modem drivers, P1394 drivers, and other media-specific drivers.

3 Diffserv (DSCP) and 802.1p Marking Behavior

The host operating system marks packets on a flow to direct these packets to certain aggregate traffic handling classes in various parts of the network. By cooperating with network policy mechanisms, the host operating system can expect to use marking to obtain a certain level of service (or quality guarantee) on behalf of applications. Marking functionality in Windows operating systems is explained in the following paragraphs.

QoS aware applications request a certain *service type* for a traffic flow, through the GQoS API. Available services are:

- Guaranteed service — offers high quality, quantifiable guarantees with bounded latency.

- Controlled load service — offers high quality, quantifiable guarantees to approximate the service that would be provided by a lightly loaded network.
- Qualitative service — generally offers medium quality, non-quantifiable guarantees.

The QoS SP responds by generating an RSVP signaling request to the network for the specified service type. Policy decision points (PDPs) in the network may refuse the request. If no PDP refuses the request, the QoS SP will invoke traffic control to mark packets on the corresponding data flow. Packets will be marked based on a mapping from requested service type to a corresponding DSCP or 802.1p mark.

The following default mapping is applied:

Service Type	DSCP	802.1p
Network Control	30 (6)	7
Guaranteed Service	28 (5)	5
Controlled Load	18 (3)	3
Qualitative	0 (0)	0
All other traffic	0 (0)	0

Notes:

1. The *network control* service cannot be requested via the GQoS API. It can however, be requested by network management applications making direct use of the TC API.
2. The actual DSCP is a six-bit field carrying the value indicated. Three of the six bits comprise a subset of the DSCP field, formerly referred to as the *IP Precedence field*. The equivalent IP precedence values are shown in parentheses.

There are several cases in which the default mapping may be overridden. These are described below.

- Non-conformance — applications requesting a quantifiable service will generally quantify the rate at which they expect traffic to be provided to the network. This rate is advertised to the network in the RSVP signaling messages and is considered by PDPs in the network when determining whether or not an RSVP request is admissible. If, for any reason, traffic control delivers packets at a rate exceeding the requested rate, these packets are considered *non-conforming* and are not entitled to the requested service level. Traffic control may therefore mark these packets accordingly.
- Custom static mappings — the network administrator may define alternate mappings for both conforming and non-conforming packets. These mappings may be individually configured in each host's registry. Alternatively, a custom mapping may be installed across a set of hosts by downloading it from the Active Directory.
- Per-flow network override — extensions to the RSVP protocol define objects (*DCLASS* and *TCLASS*) that may be provided by PDPs in response to signaling messages from hosts. These objects direct the host to use a specific DSCP or 802.1p marking (respectively) for traffic on the signaled flow.

The mechanism described above precludes Winsock2 applications from directly marking the DSCP or 802.1p field of transmitted packets. This may be considered a regression in comparison to legacy operating systems that allow applications to mark the DSCP directly. The indirection of packet marking via the QoS SP is important for several reasons:

1. Neither the DSCP nor 802.1p marks can be considered to have well-known meaning. Different networks may be provisioned to interpret these marks differently and to provide different aggregate traffic handling mechanisms in response to specific marks. Instead of requiring each application to understand the meaning of specific markings on specific networks, the QoS SP provides the mapping from well-understood service types to the appropriate mappings.
2. Marking is considered *greedy* behavior in the sense that marked packets may gain access to resources at the expense of unmarked traffic. In keeping with the objective of giving the network administrator control over network resources, packet marking must be subjugated to network policies. This is achieved by prohibiting the QoS SP from marking packets without the approval of PDPs in the affected network path.

By marking based on the abstract service requested and the results of signaling, the host is able to integrate RSVP, 802.1p and differentiated services and to provide a single, unified QoS API to applications.

There are two cases in which marking may be applied without being subjected to policy. In one case, the network administrator may configure hosts to enable legacy mode DSCP marking under direct control of Winsock2 applications. In the other case, network management applications, having administrator privileges, may directly invoke the TC API, bypassing QoS SP policy control. In both cases, the network administrator is responsible for resolving contention for network resources between traffic that is subjected to signaling-based policy controls and traffic that is not.

Note that hosts on which traffic control functionality is enabled will always mark packets in accordance with the rules above. So long as policy mechanisms are not defeated, hosts will not be able to steal network service as a result of host marking. In any case, network service providers are expected to police submitted traffic to prevent abuse of network resources by their customers. Furthermore, routers and switches in the network may always *re-mark* submitted packets. Whether and to what degree hosts are “trusted” to mark packets is a policy decision that must be made by the network administrator. One of the benefits of host marking is its scalability.

4 The Subnet Bandwidth Manager and Admission Control Service

The *Subnet Bandwidth Manager* (SBM) and the *Admission Control Service* (ACS) are Microsoft's implementation of PEP/PDP functionality. These use the Active Directory™ service as a policy data store and single point of administration. In order to act as a PEP/PDP in a QoS-enabled network, it is necessary to intercept RSVP signaling messages. Since these messages follow the data path through the network, routers or switches are the natural platforms for PEP/PDP functionality. Microsoft offers only limited router functionality. Therefore, its PEP/PDP functionality can be offered only in limited environments. Specifically, Microsoft offers PEP/PDP functionality in shared networks, based on the SBM (as described below) and in ACS/RRAS configurations (in which RRAS serves as a WAN gateway router). In other environments, industry standard router or switch-based PEP/PDPs may be employed without loss of functionality.

4.1 The Subnet Bandwidth Manager

The SBM protocol defined by the IETF extends RSVP to be useful in shared media subnetworks. In shared media subnets, there is no obvious admission control agent accountable for the shared resources. The SBM protocol defines how agents in the subnetwork elect a *Designated SBM* (DSBM) to be accountable for the shared resources of the subnet. Hosts or routers sending traffic into a shared subnet must detect the presence of a DSBM and direct their RSVP signaling messages to the DSBM as admission control agent for the subnet. This is *SBM client* functionality.

Microsoft's ACS is a service that combines the resource based admission control functionality of an SBM with policy-based admission control. The ACS leverages the fact that the DSBM (by advertisement of its presence on a shared subnet) is able to insert itself into the RSVP signaling path and, therefore, act as an admission control agent. As such, the combination of the SBM and the ACS can be used to provide PEP/PDP functionality on shared subnetworks. Note that the combination does not provide strict PEP functionality, as data traffic does not pass through it. Instead, it exerts policy control indirectly, by admitting or rejecting RSVP signaling requests.

4.2 The Local Policy Module and Extensibility

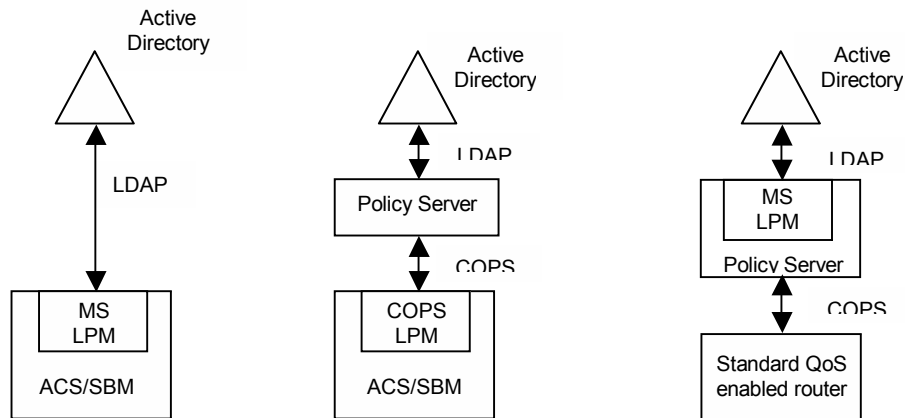
RSVP messages that are intercepted by the SBM are handed off for policy processing by a *Local Policy Module* (LPM). Microsoft's LPM simply extracts the policy-related objects from the RSVP message, applies the appropriate Kerberos processing to the user ID, and compares the requesting user ID, and the

resources requested, against privileges configured in the Active Directory. Based on the results of the comparison, the RSVP request is either admitted or rejected by the ACS. The interface between the SBM and the LPM is an open interface — the *LPM API*. Third party ISVs may use this interface to install alternate policy modules in the ACS. These policy modules may use intermediate, third party policy servers rather than accessing the Active Directory directly.

The extensibility described in the previous paragraph enables third parties to use the ACS to apply policies against their policy servers. In this model, the ACS is acting as a PEP. As explained previously, there are many scenarios in which it may be preferable to employ third party routers or switches as PEPs. Third party PEPs commonly use the Common Open Policy Service (COPS) protocol to outsource policy decisions to a PDP, which in turn uses the Active Directory as the policy data store. In these scenarios, additional extensibility is provided by allowing Microsoft's LPM to be run on third party policy servers. This mode of operation enables the policy server to readily parse Microsoft's Active Directory resident QoS schema.

4.3 Variations of the ACS

The following diagram illustrates variations of the SBM/ACS described previously.



The leftmost example illustrates a Windows 2000 server, on which the SBM/ACS service is enabled. The ACS uses the standard Microsoft LPM that uses LDAP to directly access the active directory.

The center example shows the same SBM platform, however, the Microsoft LPM has been replaced with a third party COPS LPM. The COPS LPM uses a third party policy server that may or may not use the Active Directory as its data store.

Finally, the rightmost example illustrates an industry standard router. The router uses COPS to offload the policy decision to a third party policy server. In this example, the third party policy server uses Microsoft's LPM to parse the Active Directory QoS schema.

5 The Active Directory QoS Schema

Microsoft's LPM may be used by the ACS or third party policy servers to parse the Active Directory QoS schema. This schema, together with its MMC UI, enables network administrators to use the Active Directory as a central point of control to effect signaled QoS policy for *quantitative* applications. The UI and schema enable the network administrator to specify both per subnet and enterprise-wide quantifiable resource limits for specific users or groups of users. Resource limits can be specified separately for the guaranteed and controlled load service types. At this time, Microsoft does not provide a schema to enable policy control based on user ID or application. Similarly, it does not directly support the return of DCLASS or TCLASS objects from the ACS to the host, for the purpose of overriding default mappings.

This functionality is currently available from several third party PEP/PDP vendors and will be available in the ACS and Active Directory in the subsequent release of Windows 2000.

6 Summary

Windows hosts provide a broad range of QoS functionality, including signaling, policy, marking, and traffic shaping. Host functionality integrates marking and shaping behavior with signaling and policy and presents a unified, mechanism-independent API to applications. In addition, traffic control is directly accessible to network management applications. RSVP signaling is available on Windows 2000 and Windows 98. Traffic control (with the exception of limited DSCP marking) is available only on Windows 2000.

6.1 For More Information

For the latest information on Windows 2000 Server, check out our Web site at <http://www.microsoft.com/windows/server> and the Windows NT Server Forum on MSN™ at <http://computingcentral.msn.com/topics/windowsnt>.