

The Ultimate ONT Study Package

Chris Bryant, CCIE #12933

<http://www.thebryantadvantage.com>

[Back To Index](#)

AutoQoS And Configuring QoS With Security Device Manager (SDM)

Overview

[Introduction To AutoQoS](#)

[AutoQoS Prerequisites](#)

[AutoQoS For VoIP](#)

[AutoQoS For The Enterprise](#)

[Using SDM To Generate QoS Policies](#)

["Hot Spots And Gotchas"](#)

We've got three methods of configuring and applying QoS:

- Legacy CLI ("legacy" = "old")
- MQC ("Modular QoS CLI")
- AutoQoS ("Automated / Automatic QoS")

Throughout the course, we've been creating ACLs, calling those ACLs with class maps, writing policy maps that in turn call the class maps, and then finally we've been applying the policy maps. You may *think* we were using the CLI, but in truth we were using the MQC.

Cisco's website defines the MQC as "a CLI structure used to create traffic policies and to attach these policies to interfaces". You won't notice a difference from the regular CLI, frankly, but keep that MQC in mind on exam day.

AutoQoS actually discovers what applications are running on your network - with the help of NBAR - and creates the appropriate QoS deployment.

AutoQoS is Cisco's attempt to make configuring QoS easier in today's complex networks. Several of their documents mention a cost savings as

well, since this app makes it possible for someone without in-depth knowledge of QoS to successfully deploy QoS in their network. The phrase "without calling a consulting firm and paying a consultant" is left unspoken.

Frankly, some control freaks (like me) have a hard time letting go of manually configuring QoS - that can be the hardest part of running AutoQoS! But once you work with AutoQoS, you'll wonder how you did without it.

It's hard for *anyone* to think of everything, even a network admin, but look at just a partial list of what AutoQoS will take care of for us.

On The WAN Side

- Autoconfiguration of LLQ for voice traffic, as well as guaranteeing a certain level of bandwidth to that voice traffic.
- Dynamically configures and enables RTP Header Compression (cRTP), as well as link fragmentation and interleaving (LFI)
- Autoconfigures traffic shaping as needed in compliance with Cisco best practice
- Overall, supports Frame, ATP, PPP, and HDLC - but there *are* certain interface types that do not run AutoQoS.

So on the WAN side, AutoQoS can enable LLQ and its corresponding priority queue, cRTP, and LFI. Pretty good deal!

And on the LAN Side....

- Establishes trust boundaries as needed, especially at the IP Phone
- Uses LLQ for voice traffic and WRR for data traffic
- Dynamically resizes queues and changes queue weights as deemed necessary

What interface types support AutoQoS?

- Frame Relay and ATM PTP subinterfaces
- Serial interfaces running PPP or HDLC
- ATM-to-Frame transitional links
- ATM PVCs

We're not done with the lists yet - hang in there. We've got quite a few prerequisites to concern ourselves with...

AutoQoS Prerequisites:

For WAN interfaces, the following must be configured:

- An IP address must be configured on low-speed interfaces (768 kbps

- or slower)
- *bandwidth* command is required on interfaces on both ends of the link
- CEF must be enabled at the interface level (or in the case of ATM, on the PVC itself)
- NBAR will be used to identify applications and traffic

Here's what must NOT be configured:

- QoS policies must be removed from the interface(s) that will run AutoQoS
- AutoQoS can't be implemented on/to a Frame Relay DLCI if a map class is already configured on that DLCI
- Another DLCI-related rule: If one subinterface already has a Frame DLCI, you can't configure AutoQoS on a different subinterface.

Hang in there, it gets easier. :)

There are two flavors of AutoQoS:

AutoQoS VoIP - runs on routers and Cat switches, including the popular Cisco home lab 2950 (Enhanced image required)

AutoQoS Enterprise - runs only on routers. Consists of two stages, *Autodiscovery* (using NBAR or DCSP markings) and *Template Generation And Installation*.

I'll try not to mention this again for at least five minutes, but CEF must be enabled for either version of AutoQoS to work correctly.

Let's take a look at the VoIP version of AutoQoS.

AutoQoS For VoIP

If you support a small or not-so-small network, this feature is for you! Coming up with a QoS solution for your VoIP network is time-consuming, and that's one thing many network admins just can't spare. AutoQoS for VoIP can really come in handy for networks who just do not have the trained personnel and/or budget to come up with a QoS scheme on their own.

Cisco's website lists quite a few benefits for AutoQoS VoIP, including:

- Establishment of trust boundaries, especially with Cisco IP Phones and access ports
- Autoconfiguration of strict priority queuing for VoIP traffic (LLQ) and weighted round robin (WRR) transmission of data
- Dynamic modification of queue sizes and weights as needed
- Automatic configuration of QoS parameters based on both Cisco best practices and input from previous installations of Cisco Unified

Communications

AutoQoS for VoIP has two requirements and about 30 warnings. Let's look at the requirements first:

- An IP address must be configured on the interface or subinterface
- The *bandwidth* command must be configured on those interfaces

The *bandwidth* command's value will be used by the AutoQoS process, so we need to double-check that value's accuracy.

AutoQoS for VoIP will create global templates for ACLs, class maps, and policy maps - everything we need for QoS! NBAR is used to classify the voice traffic and assign an appropriate DSCP value.

You also have the option to trust previously-applied DSCP values on incoming packets.

As always with AutoQoS, any preexisting service policy *must* be removed from any interface that will be running AutoQoS for VoIP.

AutoQoS VoIP runs on these interface types:

- Serial interfaces running HDLC or PPP
- Frame Relay DLCIs on PTP subinterfaces

There are some other Frame Relay "gotchas" that you might run into:

- If a Frame Relay DLCI is configured with a map class, it can't then be configured with AutoQoS.
- AutoQoS and virtual templates do not play well together - if the DLCI is configured with a virtual template, you can't run AutoQoS on that DLCI.

Man, that's a lot of rules! :) The good news is that the simple command *auto qos voip* enables this feature. We have one interesting option to consider as well, shown here with IOS Help:

```
R1(config)#int s0/1
R1(config-if)#auto qos ?
  voip  Configure AutoQoS for VoIP
  <cr>
```

```
R1(config-if)#auto qos voip ?
  trust Trust the DSCP marking
  <cr>
```

```
R1(config-if)#auto qos voip
```

If you use the *auto qos voip* command by itself, NBAR will be used to discover traffic. If you choose the *trust* option, the incoming DSCP values

are trusted and NBAR is not used to discover and classify the traffic.

AutoQoS For The Enterprise

Enterprise is a two-step process, with the commands shown in parenthesis:

- *Autodiscovery* ("*auto discovery qos*")
- *MQC-Based Policy Generation And Application* ("*auto qos*")

The order certainly makes sense, as we need to discover what applications and traffic we're running before developing a QoS scheme!

Not all interfaces can run Enterprise. Here are the ones that can:

- Frame DLCIs, PTP subinterfaces only (no map classes)
- ATM PVCs
- Serial interfaces running PPP or HDLC

Other interface prerequisites:

- The bandwidth command must be configured along with an IP address on "low-speed" Serial interfaces (768 kbps or slower)
- CEF must be enabled

The autodiscovery phase is put into action with the *auto discovery qos* command. (It's not the *auto qos* command - that comes later.)

We have the same trust option with Enterprise as we did with VoIP:

```
R1(config)#int s0/0
R1(config-if)#auto ?
  discovery  Configure Auto Discovery
  qos       Configure AutoQoS

R1(config-if)#auto discovery ?
  qos       Configure Auto Discovery for QoS

R1(config-if)#auto discovery qos ?
  trust     Trust the DSCP marking
<cr>

R1(config-if)#auto discovery qos
```

Note the option "trust" for this command. If you use the basic *auto discovery qos* command, NBAR will be used to discover traffic; if you use the *trust* option, traffic will be classified by DSCP.

Regardless of which option you choose, be prepared for the router to "pause" for 15 seconds or so after you configure this command. The prompt for the next line will take that long to appear, and if this is the first

time you've used the command, it's easy to think that the router is "stuck".

Now that we've got discovery running, we'll configure the auto qos command on both R1 and R2.

```
R1(config-if)#auto qos
```

```
R2(config-if)#auto qos
```

We'll run the *show auto discovery qos* command to view the results. This command also shows you how long the discovery process has been running.

The output of this command is *huge*, so we'll start with the class information you'll see at the top of the output. Obviously, you'll need to run Autodiscovery for longer than four minutes in a production network!

```
R1#show auto discovery qos
Serial0/1/0
AutoQoS Discovery enabled for applications
Discovery up time: 4 minutes, 29 seconds
AutoQoS Class information:
Class Voice:
  Recommended Minimum Bandwidth: 43 Kbps/5% (PeakRate)
  Detected applications and data:
  Application/           AverageRate           PeakRate           Total
  Protocol              (kbps/%)            (kbps/%)          (bytes)
  -----
  rtp audio             6/<1                 43/5              226032
Class Interactive Video:
  No data found.
Class Signaling:
  Recommended Minimum Bandwidth: 11 Kbps/1% (AverageRate)
  Detected applications and data:
  Application/           AverageRate           PeakRate           Total
  Protocol              (kbps/%)            (kbps/%)          (bytes)
  -----
  h323                  11/1                 22/2              382727
  rtcp                  0/0                  0/0               2496
Class Streaming Video:
  No data found.
Class Transactional:
  Recommended Minimum Bandwidth: 62 Kbps/8% (AverageRate)
  Detected applications and data:
  Application/           AverageRate           PeakRate           Total
  Protocol              (kbps/%)            (kbps/%)          (bytes)
  -----
  sqlnet                61/7                 116/15            2054083
  citrix                1/<1                  5/<1               63552
Class Bulk:
  Recommended Minimum Bandwidth: 42 Kbps/5% (AverageRate)
  Detected applications and data:
  Application/           AverageRate           PeakRate           Total
  Protocol              (kbps/%)            (kbps/%)          (bytes)
  -----
  exchange              24/3                 47/6              817004
  ftp                   18/2                 35/4              625428
Class Scavenger:
  Recommended Minimum Bandwidth: 8 Kbps (AverageRate)/1% (fixed)
```

```

Detected applications and data:
Application/      AverageRate      PeakRate         Total
Protocol         (kbps/%)        (kbps/%)        (bytes)
-----
kazaa2           8/1              17/2             284312
Class Management:
Recommended Minimum Bandwidth: 14 Kbps/1% (AverageRate)
Detected applications and data:
Application/      AverageRate      PeakRate         Total
Protocol         (kbps/%)        (kbps/%)        (bytes)
-----
ldap             14/1             27/3             473712
Class Routing:
Recommended Minimum Bandwidth: 0 Kbps/0% (AverageRate)
Detected applications and data:
Application/      AverageRate      PeakRate         Total
Protocol         (kbps/%)        (kbps/%)        (bytes)
-----
icmp             0/0              0/0              2640
ospf             0/0              0/0              1176
Class Best Effort:
Current Bandwidth Estimation: 510 Kbps/66% (AverageRate)
Detected applications and data:
Application/      AverageRate      PeakRate         Total
Protocol         (kbps/%)        (kbps/%)        (bytes)
-----
unknowns        469/61          904/117         15797717
netbios         29/3             56/7             1006088
http            12/1             22/2             405744

```

As you can see, the protocols running in each class of traffic are displayed. For example, under *Class Routing* (the next to last category shown), you can see ICMP and OSPF.

The next information is the suggested AutoQoS policy, with the class maps followed by the AutoQoS policy. Take note of the classes - if you're creating your own classes rather than using AutoQoS, this is a good road map to follow.

- Voice
- Signaling
- Transactional
- Bulk
- Scavenger
- Management

```

Suggested AutoQoS Policy for the current uptime:
!
class-map match-any AutoQoS-Voice-Se0/1/0
  match protocol rtp audio
!
class-map match-any AutoQoS-Signaling-Se0/1/0
  match protocol h323
  match protocol rtcp
!
class-map match-any AutoQoS-Transactional-Se0/1/0
  match protocol sqlnet
  match protocol citrix
!

```

```

class-map match-any AutoQoS-Bulk-Se0/1/0
  match protocol exchange
  match protocol ftp
!
class-map match-any AutoQoS-Scavenger-Se0/1/0
  match protocol kazaa2
!
class-map match-any AutoQoS-Management-Se0/1/0
  match protocol ldap
!
policy-map AutoQoS-Policy-Se0/1/0
  class AutoQoS-Voice-Se0/1/0
    priority percent 5
    set dscp ef
  class AutoQoS-Signaling-Se0/1/0
    bandwidth remaining percent 1
    set dscp cs3
  class AutoQoS-Transactional-Se0/1/0
    bandwidth remaining percent 8
    random-detect dscp-based
    set dscp af21
  class AutoQoS-Bulk-Se0/1/0
    bandwidth remaining percent 5
    random-detect dscp-based
    set dscp af11
  class AutoQoS-Scavenger-Se0/1/0
    bandwidth remaining percent 1
    set dscp cs1
  class AutoQoS-Management-Se0/1/0
    bandwidth remaining percent 1
    set dscp cs2
  class class-default
    fair-queue

```

We'll verify the AutoQoS configuration with *show auto qos*. Some of the same information is repeated from the earlier command - the policy maps and class maps, to be specific - but at the very bottom of the *show auto qos* output, you'll see the *service-policy* command which applies these maps.

```

R1#show auto qos
!
policy-map AutoQoS-Policy-Se0/1/0
  class AutoQoS-Voice-Se0/1/0
    priority percent 5
    set dscp ef
  class AutoQoS-Signaling-Se0/1/0
    bandwidth remaining percent 2
    set dscp cs3
  class AutoQoS-Transactional-Se0/1/0
    bandwidth remaining percent 15
    random-detect dscp-based
    set dscp af21
  class AutoQoS-Bulk-Se0/1/0
    bandwidth remaining percent 10
    random-detect dscp-based
    set dscp af11
  class AutoQoS-Scavenger-Se0/1/0
    bandwidth remaining percent 1
    set dscp cs1
  class AutoQoS-Management-Se0/1/0
    bandwidth remaining percent 3

```



```

set dscp cs2
class class-default
  fair-queue
!
class-map match-any AutoQoS-Transactional-Se0/1/0
  match protocol sqlnet
  match protocol citrix
!
class-map match-any AutoQoS-Voice-Se0/1/0
  match protocol rtp audio
!
class-map match-any AutoQoS-Scavenger-Se0/1/0
  match protocol kazaa2
!
class-map match-any AutoQoS-Signaling-Se0/1/0
  match protocol h323
  match protocol rtcp
!
class-map match-any AutoQoS-Bulk-Se0/1/0
  match protocol exchange
  match protocol ftp
!
class-map match-any AutoQoS-Management-Se0/1/0
  match protocol ldap
!
Serial0/1/0 -
!
interface Serial0/1/0
  ip address 10.2.1.1 255.255.255.0

  encapsulation ppp
  bandwidth 768
  service-policy output AutoQoS-Policy-Se0/1/0
  ip rtp header-compression iphc-format

```

And that's it! We now have an AutoQoS policy configured and applied, all done with just the auto discovery qos and auto qos commands!

Enterprise AutoQoS Benefits

- It's easier and cheaper (theoretically)
- Resulting QoS deployment can be fine-tuned at the MQC level
- As with AutoQoS for VoIP, Enterprise will implement LLQ, cRTP, and LFI

AutoQoS Does NOT Dynamically Adapt To Bandwidth Change

Can changes be made to settings such as *bandwidth* after configuring AutoQoS?

Yes, BUT... AutoQoS does not dynamically adapt to such a change. AutoQoS must be disabled and then reenabled for the new settings to be used by AutoQoS.

That change does not make the current AutoQoS deployment invalid, but to reflect the new *bandwidth* value, the process must be stopped and restarted.

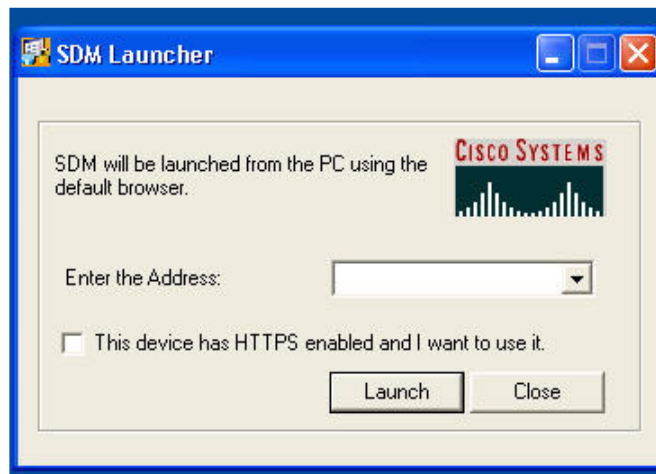
Configuring QoS With SDM

Cisco's Security Device Manager (SDM) is a powerful GUI-based tool for configuring everything from VPNs to QoS - and it's QoS we'll configure right now with SDM.

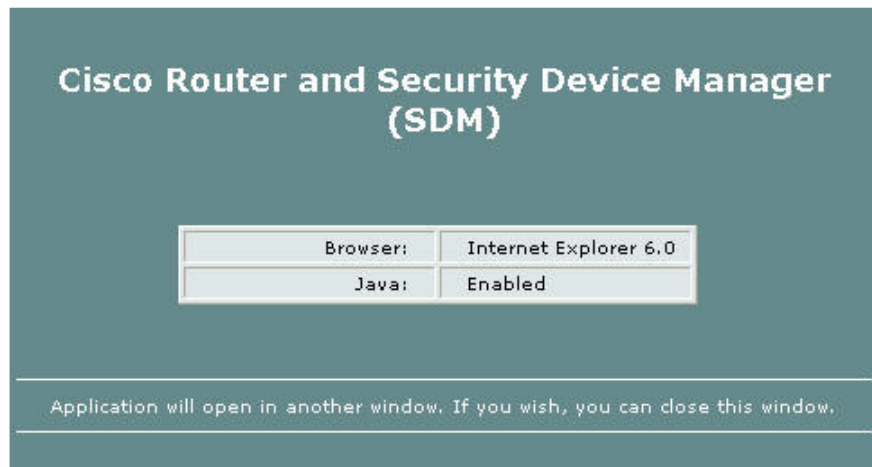
Once SDM is installed, a desktop icon will be available for you to click and launch the program.



You'll then be prompted for the IP address of the router you wish to configure. The option to connect via Secure HTTP is also available, but is not checked by default.



Two separate browsers will open, and one will display this message:



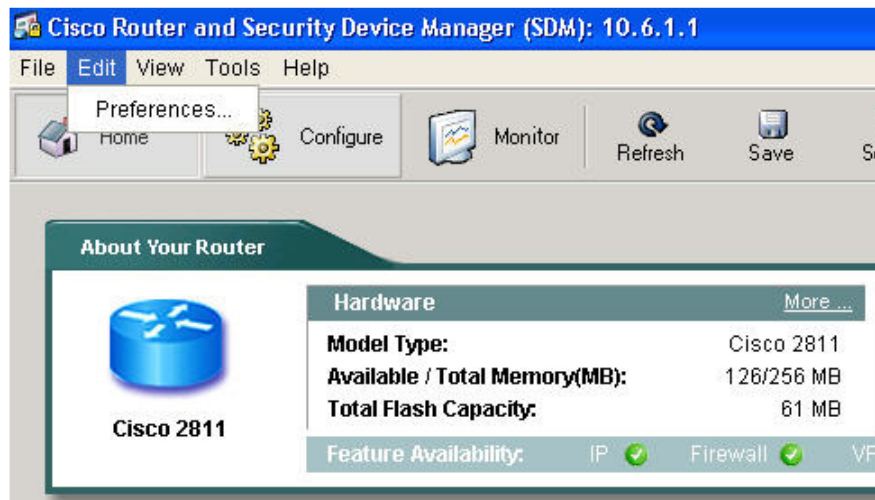
The second will display this message:



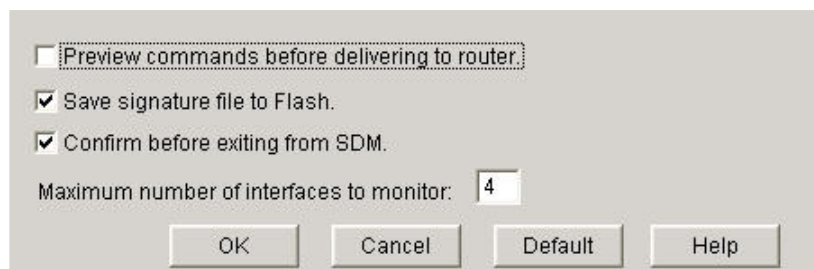
Did you notice the main difference? It's okay to close the first one, but if you close the second window, SDM will shut down. So don't close this one until you're done! :)

SDM opens to the Home window. This window gives you some basic information regarding your router, along with an overview of the current configuration.

There's one SDM default I prefer to change, so we'll click *Edit* and then go to *Preferences* on the Home window.



Here are the three preferences, shown with their default settings:

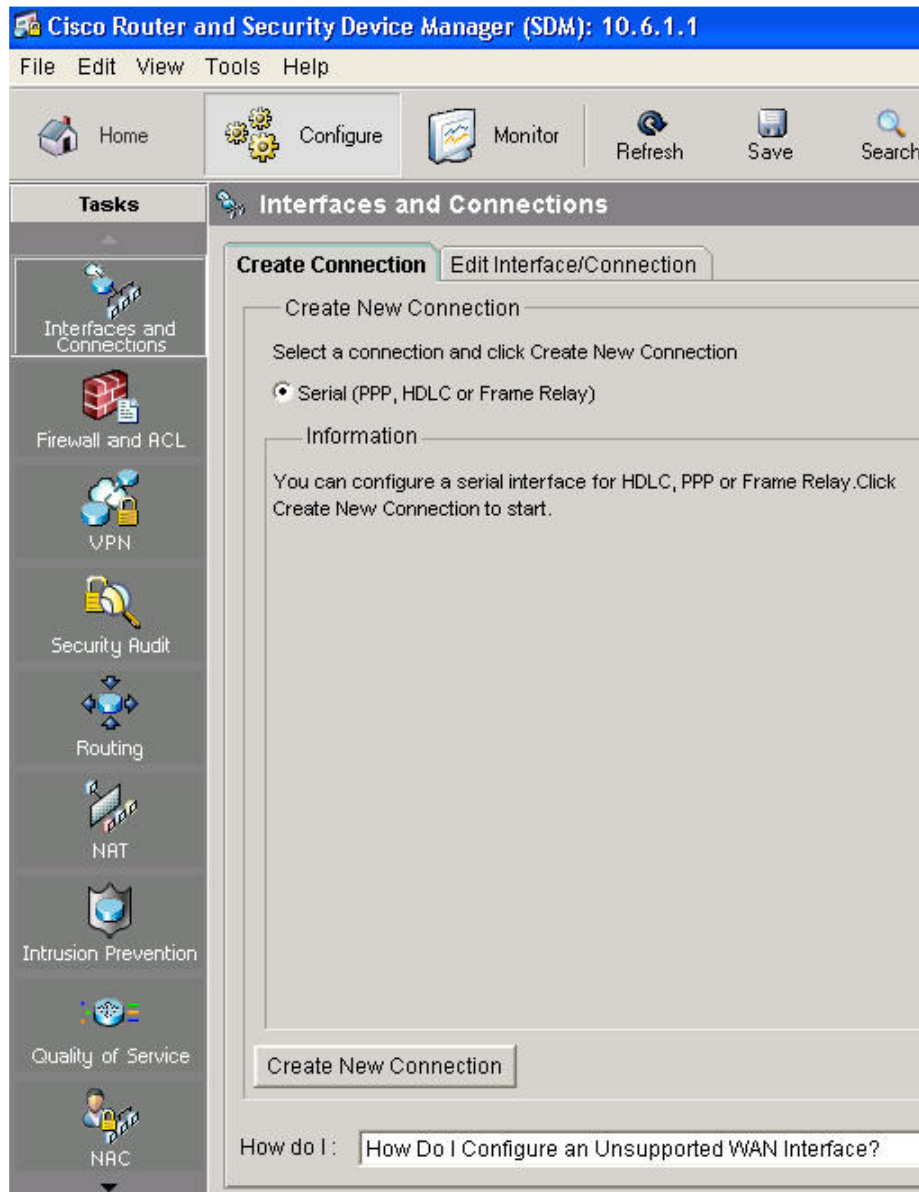


I personally like to preview the commands before SDM configures the router, so I'll check that box and move on. That's strictly personal preference and not something you have to do, nor is it a "best practice".

We'll click the *Configuration* button next.



Interfaces and Connections will be chosen on the next screen by default, but that's not the one we want. We'll choose the *Quality of Service* button instead. Before we move on, here's the Configure screen:



Note the "How do I" selection at the bottom of the screen. Every SDM selection in the Configure section has this option at the bottom, and if you're unsure of how to carry out a task, this is very helpful!

I click the QoS button, and here's the resulting screen. The SDM screens give excellent descriptions of what you're about to do.

Quality of Service


Create QoS Policy | Edit QoS Policy

SDM can guide you in configuring a basic Quality of Service (QoS) policy for outgoing traffic on WAN interfaces and IPsec tunnels.

SDM creates a Low Latency Queuing (LLQ) service policy with its associated classes. The service policy is created by allocating proportional bandwidth on the WAN/IPsec interfaces, and bandwidth you specify for the constituent classes.

The service policy is then associated with the WAN or IPsec interface you select.

Use Case Scenario



Launch QoS Wizard

I'll click *Launch QoS Wizard*, and here's the result!

QoS Wizard

QoS Wizard guides you in configuring a default Quality of Service (QoS) policy for your WAN interfaces.

SDM, by default, would create a QoS policy to handle 2 main types of traffic:

1) Real-Time

Under this traffic, SDM considers VoIP and signaling packets.

2) Business-Critical

Under this traffic, SDM considers 3 sub-categories of traffics -

a) Transactional - handles packets meant for ERP/Database, Interactive Sessions, Enterprise Applications.

b) Management - handles packets meant for Network Management.

c) Routing - handles packets meant for Routing and Signaling.

A few notes of interest....

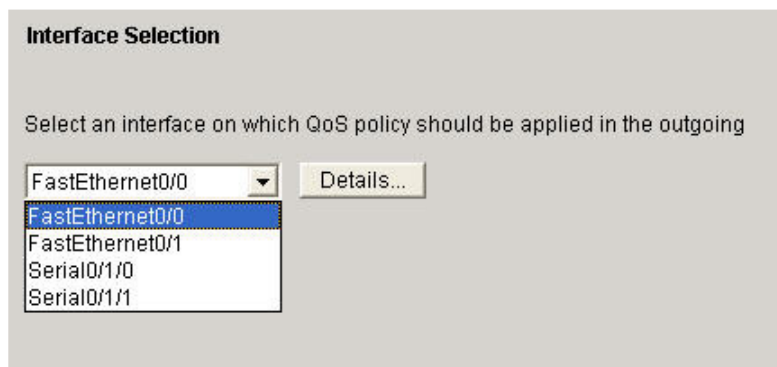
The QoS policy created will handle 2 main types of traffic, and since not all traffic will fall under these two categories, we effectively have three different types of traffic:

- Real-Time (VoIP *and* signaling protocols)
- Business-critical
- Best Effort (anything that doesn't match the first two)

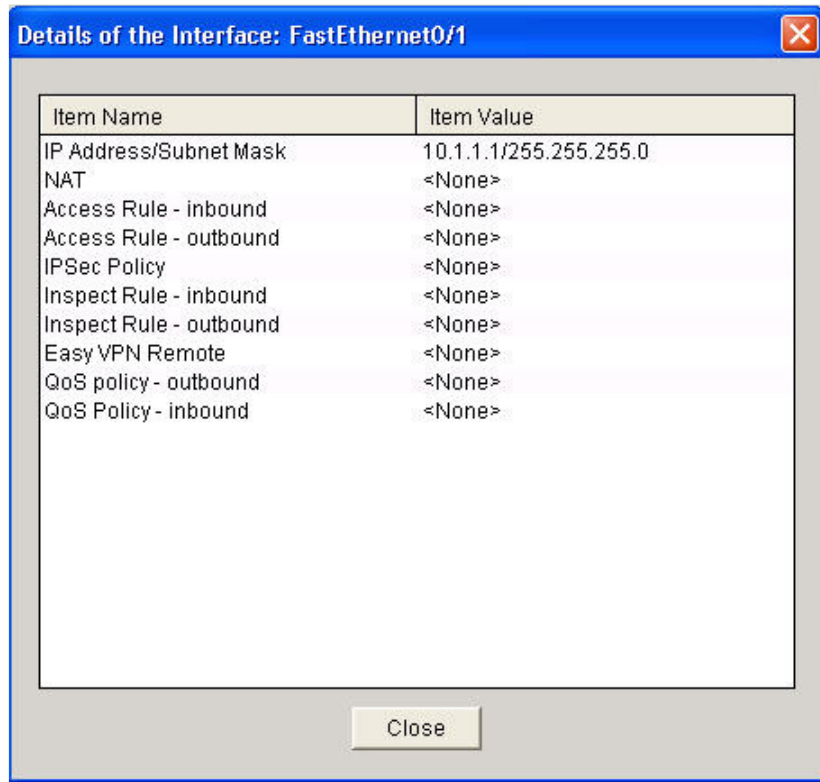
We have three subclasses of Business-Critical traffic:

- Transactional
- Management
- Routing

There's a *Next* button not shown in the above screen - I'll click that and we'll move on.



The drop-down box lists the interface upon which we will apply this QoS policy for outgoing traffic. Once you select an interface, you can click *Details* for additional information about that particular interface, and here's the result.



I'll click *Close*, then *Next*, and we move to the next screen!

QoS Policy Generation

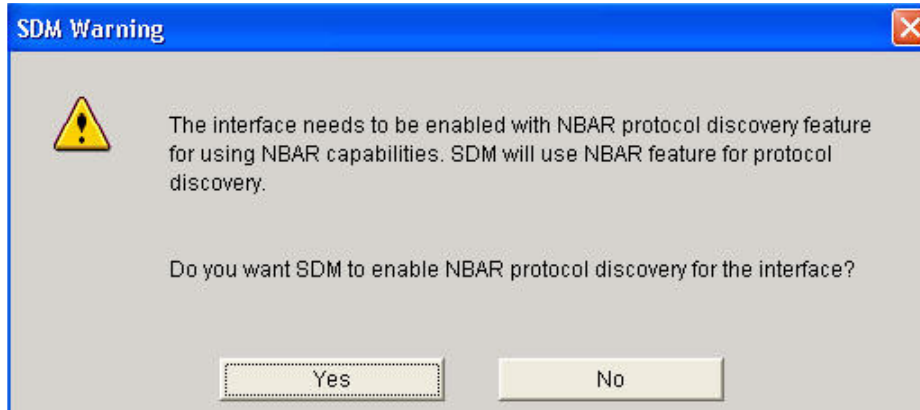
SDM will create a QoS policy to provide quality of service to 2 types of traffic:

- 1) Real-Time Traffic :- SDM will create 2 QoS classes to handle VoIP and voice signaling packets.
- 2) Business-Critical Traffic :- SDM will create 3 QoS classes to handle packets which are important for a typical corporate environment. Some of the protocols included in this traffic category are citrix, sqlnet, notes, LDAP, and secure LDAP. Routing protocols in this category include BGP, EGP, EIGRP AND RIP.

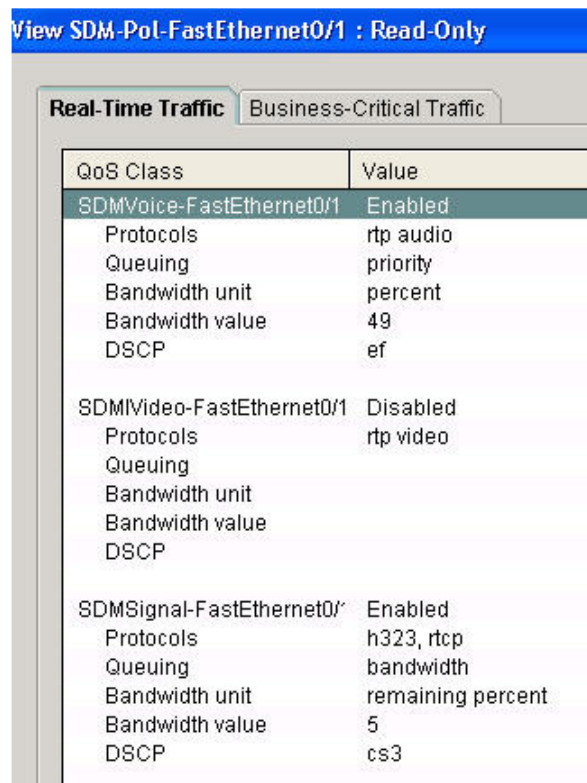
Bandwidth Allocation

| Type of Traffic | Bandwidth in % | kbps value |
|----------------------------|----------------------------------|-------------------------------------|
| Real Time (Voice, Video) : | <input type="text" value="72"/> | <input type="text" value="72000"/> |
| Business-Critical : | <input type="text" value="2"/> | <input type="text" value="2000"/> |
| Best-Effort : | <input type="text" value="26"/> | <input type="text" value="26000"/> |
| <hr/> | | |
| Total Bandwidth : | <input type="text" value="100"/> | <input type="text" value="100000"/> |

Now it gets interesting. We have to specify the percentage of bandwidth we want to be assigned to Realtime, Business-Critical, and Best-Effort traffic. I'll assign 50% of our bw to Realtime, and split the rest equally. After that, I'll click *View Details*, and this is the result.

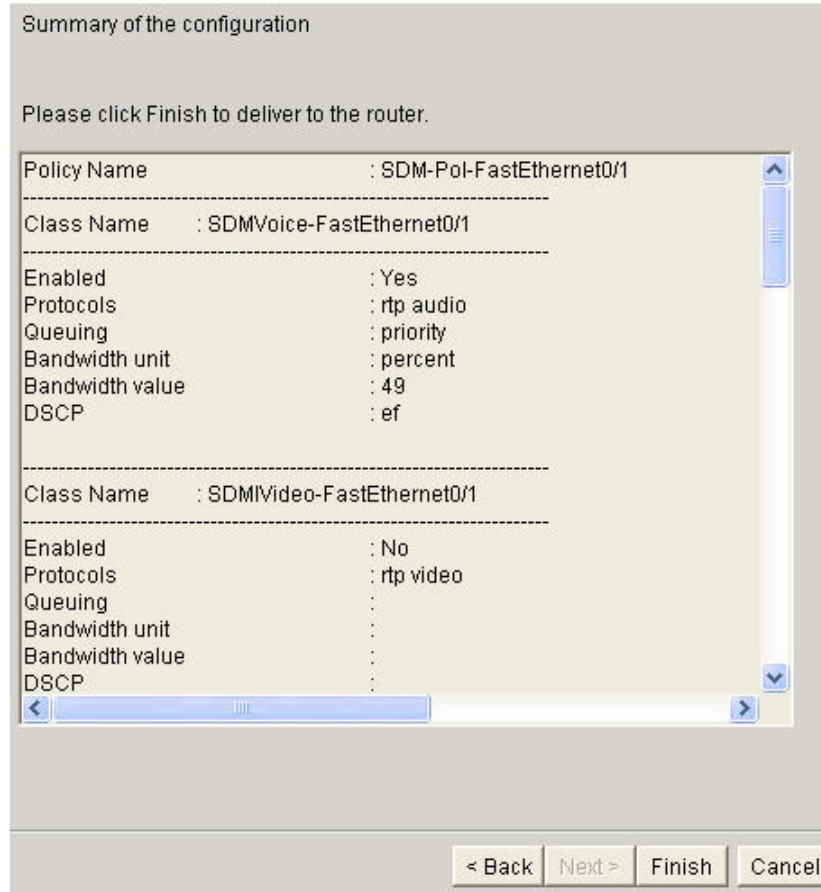


I'll click *Yes*, and in a few minutes we get a report on the traffic NBAR has discovered. I will not show all of the protocols, but here's a sample of what you'll see. Note that Realtime traffic shows by default and that there is a separate tab for Business-Critical traffic. RTP Audio traffic will be marked with a DSCP value of EF (Expedited Forwarding).

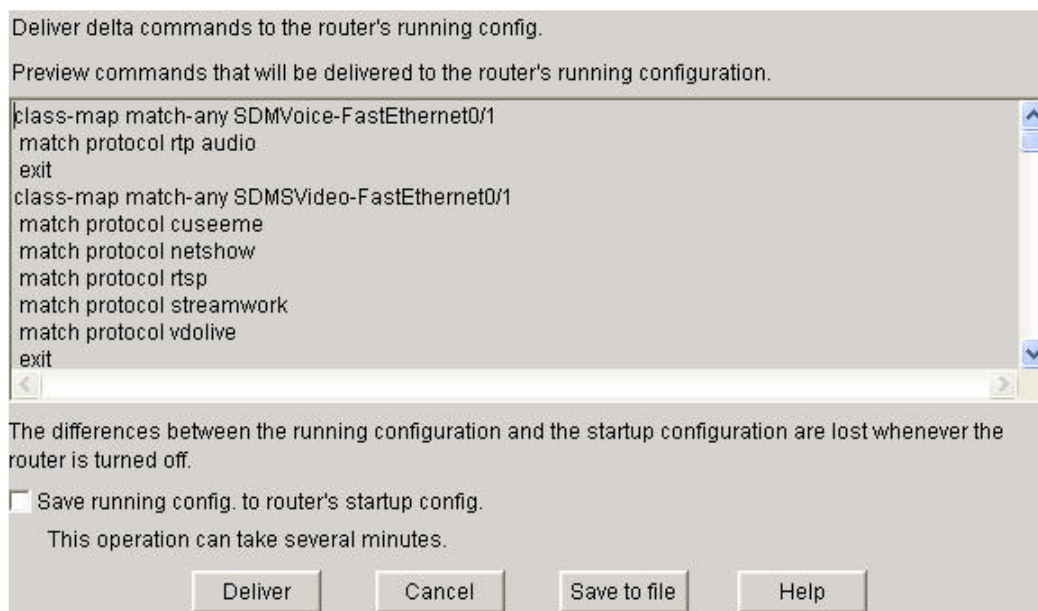
A screenshot of a network configuration window titled "View SDM-Pol-FastEthernet0/1 : Read-Only". It has two tabs: "Real-Time Traffic" (selected) and "Business-Critical Traffic". Below the tabs is a table with two columns: "QoS Class" and "Value".

| QoS Class | Value |
|---------------------------|-------------------|
| SDMVoice-FastEthernet0/1 | Enabled |
| Protocols | rtp audio |
| Queuing | priority |
| Bandwidth unit | percent |
| Bandwidth value | 49 |
| DSCP | ef |
| SDMVideo-FastEthernet0/1 | Disabled |
| Protocols | rtp video |
| Queuing | |
| Bandwidth unit | |
| Bandwidth value | |
| DSCP | |
| SDMSignal-FastEthernet0/1 | Enabled |
| Protocols | h323, rtcp |
| Queuing | bandwidth |
| Bandwidth unit | remaining percent |
| Bandwidth value | 5 |
| DSCP | cs3 |

After closing that window and clicking *Next*, we're shown a summary of the configuration that will be delivered to the router.

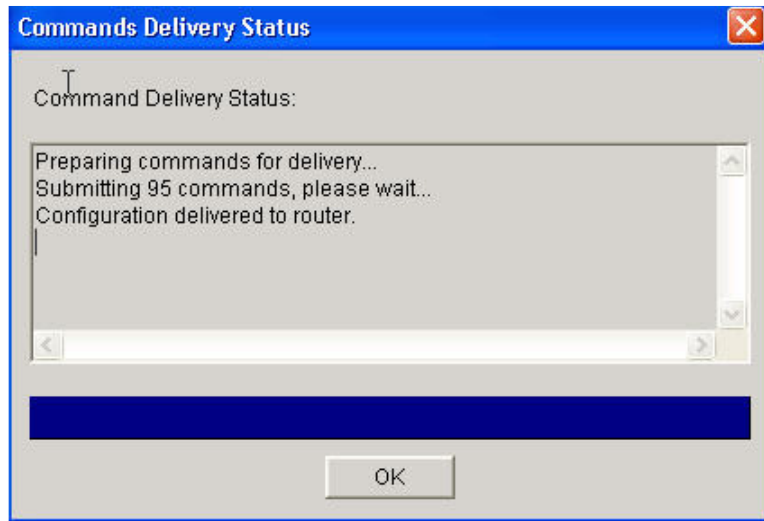


If we had left the Preferences at their default, the configuration would be delivered to the router when we click *Finish*. Since I changed one preference and requested the configuration be shown before delivery, we get the following window instead.



The scroll bar on the right side can be used to browse the config if you like. Note the option to save the running config to the startup config, and that it is not selected by default. Personally, I've seen that save fail more than once, so I like to do that myself. Again, it's a personal preference.

When I click *Deliver*, that's just what happens - the commands are delivered to the router.

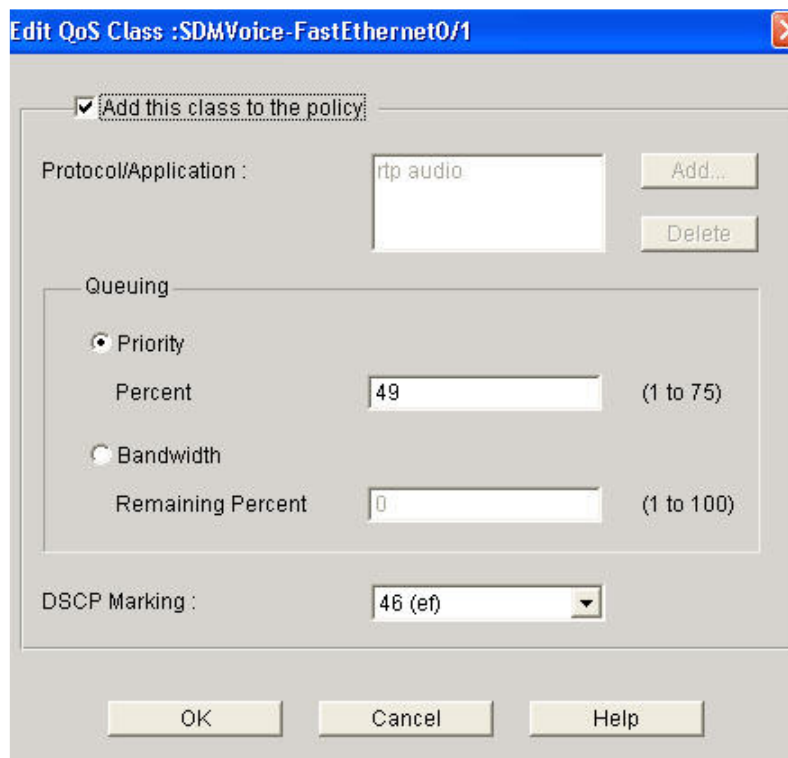


As the commands are delivered, that solid bar shown above the "OK" button will scroll back and forth. It can take a few minutes to deliver the configuration, but in this case it took only a few seconds. Note that 95 commands were delivered!

I'll click *OK*, and we're taken to the *Edit QoS Policy* window. Note the queueing schemes - priority and bandwidth, with priority next to Voice.

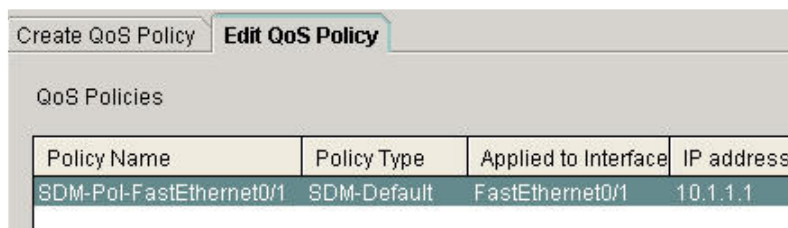
| Traffic Type | Class Name | Enabled | Protocols | Queueing | Percent |
|-------------------|---------------------------|---------|----------------------|-----------|---------|
| Real-Time | SDMVoice-FastEthernet0/1 | ✓ | rtp audio | priority | 49 |
| Real-Time | SDMSVideo-FastEthernet0/1 | ✗ | cuseeme, netsh | | |
| Real-Time | SDMSignal-FastEthernet0/1 | ✓ | h323, rtcp | bandwidth | |
| Real-Time | SDMIVideo-FastEthernet0/1 | ✗ | rtp video | | |
| Business-Critical | SDMTrans-FastEthernet0/1 | ✓ | citrix, finger, note | bandwidth | |
| Business-Critical | SDMManage-FastEthernet0/1 | ✓ | dhcp, dns, imap | bandwidth | |
| Business-Critical | SDMRout-FastEthernet0/1 | ✓ | bgp, egp, eigrp, | bandwidth | |

To edit any of the QoS settings, just highlight the feature you want to edit and click - what else? - the grey *Edit* button. Since the SDMVoice class was highlighted in the previous illustration, I just clicked Edit and up pops the following window:



You can also use SDM to see the DSCP value that will be assigned to different traffic classes, as well as the LLQ *priority* and *bandwidth* queues.

At the top of the *Edit QoS Policy* window, you'll see the policies and the interfaces they've been applied to. Here we only have one policy, but in situations where you have multiple policies, make sure you're reading and editing the correct one for the interface you're working with.



At the bottom of the screen, you'll see the following information. (To allow this to fit into one screen, I've narrowed the *Enabled* and *Remaining Percentage* fields, which are not the ones we're concerned with right now.)

| QoS Policy Details | | Bandwidth Allocation - | | Real-Time: 49% | | Business-Critical | |
|--------------------|---------------------------|------------------------|-----------|----------------|----|-------------------|--|
| Traffic Type | Class Name | Protocols | Queuing | Perce | Re | DSCP | |
| Real-Time | SDMVoice-FastEthernet0/1 | rtp audio | priority | 49 | | ef | |
| Real-Time | SDMVideo-FastEthernet0/1 | cuseeme, ne | | | | | |
| Real-Time | SDMSignal-FastEthernet0/1 | h323, rtp | bandwidth | | 5 | cs3 | |
| Real-Time | SDMVideo-FastEthernet0/1 | rtp video | | | | | |
| Business-Critical | SDMTrans-FastEthernet0/1 | citrix, finger, i | bandwidth | | 79 | af21 | |
| Business-Critical | SDMManage-FastEthernet0/1 | dhcp, dns, in | bandwidth | | 8 | cs2 | |
| Business-Critical | SDMRout-FastEthernet0/1 | bgp, egp, eig | bandwidth | | 8 | cs6 | |

RTP Audio traffic will be put into the LLQ priority queue, as we'd expect. The DSCP marking for that traffic will be *ef*. You can see the other DSCP markings for various traffic types.

As with anything new, SDM takes a little getting used to - but frankly, you'll pick it up very quickly the first time you use it, whether that be in the exam room or working on a production network. Just keep your calm and keep looking around - SDM is a very intuitive GUI and you'll enjoy using it.

"Hot Spots And Gotchas"

AutoQoS will create and apply QoS templates in accordance with network traffic flows observed during the Discovery phase. AutoQoS will also establish an LLQ scheme for voice traffic while using WRR for data.

Remember the Trust Boundary discussion from another section? AutoQoS can also determine where the Trust Boundary begins, but I'd keep an eye on that personally as well.

AutoQoS is often configured at the MQC, and some consider that the best way to change values as well. That can also be done via SDM.

Sometimes it seems as though it's tougher to remember the rules and prerequisites for AutoQoS than it is to actually use it! So to review...

The *bandwidth* command should be run on the interfaces and subinterfaces that will run AutoQoS. This value is used to calculate several QoS features, including LLQ, Real-Time Protocol Header Compression, and LFI among others.

And if you change the *bandwidth* value, AutoQoS will not dynamically adjust - you need to go through the entire process again and generate new templates.

CEF must be enabled on the interfaces running AutoQoS.

You must configure an IP address on all low-bandwidth interfaces.

You must remove any previously configured QoS policies.

According to Cisco's website, the following interface types support AutoQoS:

Serial interfaces, whether they're running HDLC or PPP

Frame Relay interfaces and PTP subinterfaces (not multipoint subinterfaces)

Both low- and high-speed ATM PVCs configured on PTP subinterfaces (again, not multipoint subinterfaces)

Frame Relay-to-ATM links

AutoQoS and virtual templates do *not* work well together.

SDM AutoQoS Wizard (And Other Wizards)

We concentrated on the QoS Wizard in this section, but SDM is capable of much more when it comes to security and routing - you can configure routing protocols, NAT, VPNs, and perform security audits - and those are just *some* of SDM's features.

SDM really has three major purposes:

- Ease of configuration
- Spotting potential issues with a configuration
- Suggesting fixes and implementing them if the admin gives the go-ahead

SDM will not only create a LLQ service policy, but will create a *priority* queue (as we'd expect with LLQ) and assign a certain amount of bandwidth to the other queues.

As we saw in the lab, SDM uses NBAR to gather information about traffic flows in the network.

SDM configures QoS for both Realtime and Business-Critical traffic. All other traffic will receive "best-effort" delivery.

As we saw in the lab, SDM will create two realtime traffic subclasses (VOIP and signaling) and three business-critical traffic subclasses.

QoS Wizard

QoS Wizard guides you in configuring a default Quality of Service (QoS) policy for your WAN interfaces.

SDM, by default, would create a QoS policy to handle 2 main types of traffic:

1) Real-Time

Under this traffic, SDM considers VoIP and signaling packets.

2) Business-Critical

Under this traffic, SDM considers 3 sub-categories of traffics -

a) Transactional - handles packets meant for ERP/Database, Interactive Sessions, Enterprise Applications.

b) Management - handles packets meant for Network Management.

c) Routing - handles packets meant for Routing and Signaling.

Don't be thrown when you look at the actual configuration generated by this wizard - those Business-Critical classes will appear as SDMTrans, SDMManage, and SDMRoute. (SDM likes to put its initials on everything it does!)

Interestingly enough, SDM's Policy Generation phase will have you fill in the percentage of bandwidth you wish you assign to Realtime and Business-Critical classes, but will then automatically assign the remaining percentage to Best Effort.

QoS Policy Generation

SDM will create a QoS policy to provide quality of service to 2 types of traffic:

1) Real-Time Traffic :- SDM will create 2 QoS classes to handle VoIP and voice signaling packets.

2) Business-Critical Traffic :- SDM will create 3 QoS classes to handle packets which are important for a typical corporate environment. Some of the protocols included in this traffic category are citrix, sqlnet, notes, LDAP, and secure LDAP. Routing protocols in this category include BGP, EGP, EIGRP AND RIP.

Bandwidth Allocation

| Type of Traffic | Bandwidth in % | kbps value |
|----------------------------|----------------------------------|-------------------------------------|
| Real Time (Voice, Video) : | <input type="text" value="72"/> | <input type="text" value="72000"/> |
| Business-Critical : | <input type="text" value="2"/> | <input type="text" value="2000"/> |
| Best-Effort : | <input type="text" value="26"/> | <input type="text" value="26000"/> |
| <hr/> | | |
| Total Bandwidth : | <input type="text" value="100"/> | <input type="text" value="100000"/> |

[View Details...](#)

Okay, it's not *that* interesting, I admit. But this part of the Wizard will not leave any bandwidth unassigned.

Hey, remember that you can also run AutoQoS in the MQC! :) Let's review those commands...

auto discovery qos starts the Discovery process, using NBAR to observe and record traffic flows.

auto qos actually creates the class maps and policy maps we saw in this section and applies them to that interface.

Keep the purpose of both of those commands straight! Running those two commands in that order is almost all there is to actually running AutoQoS - it's all of the prerequisites you have to watch out for!

Both of those commands are interface-level commands.

To run AutoQoS for VoIP, use the *auto qos voip* command. AutoQoS for VoIP is an excellent way to automate the entire process of developing a QoS scheme for voice, and it's a real boon for networks whose admins may not know enough voice to do that on their own.