

## Module 3

### Quality of Services QoS/Part 2

Prepared by Dr. Manal Mizher

## Lecture 5

# جامعة الجميع الذكية

تسعى لتكون الكيان الأول الذي يتبنى مبدأ المجتمع التعليمي  
التعاوني ويقدم العلم مقابل العمل لتنمية وبناء المجتمعات.

## Network programming

Course ID: CYB615



جامعة الجميع الذكية  
EVERYONE'S SMART UNIVERSITY

## 1. QoS at Layers

(Application layer, Transport layer, Network layer, Data link layer, QoS ipv4 vs ipv6 )

## 2. ToS & DSCP

(IP Precedence, 8 bits, DSCP to Service Class Mapping, )

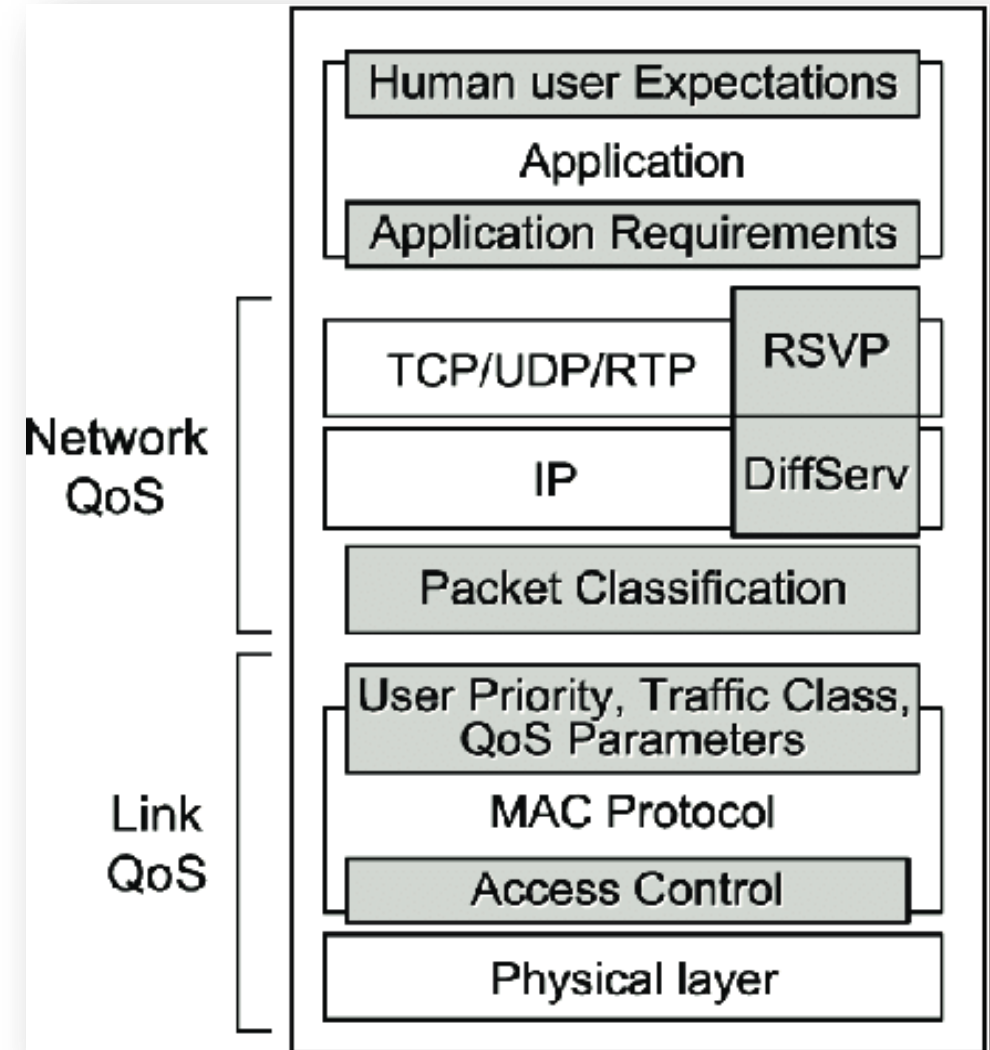
## 3. Configuration for QoS

(cmd → ping -v , )

## 4. Quick Review for :

- NetfilterQueue 1.0.0
- QoS within Sockets (*IPPROTO\_IP → IP\_TOS, QoS API in windows*  
*SOL\_SOCKET/SO\_PRIORITY,*  
*IPPROTO\_IPV6 → IPV6\_TCLASS, IPV6\_RECVTCLASS,*  
*ICMP → traffic\_class*)

- 1) QoS in the Application Layer
  - 2) QoS in the Transport Layer
  - 3) QoS in the Network Layer
  - 4) QoS in the Link Layer
- QoS could be achieved in any of the protocol layers.
  - Most existing QoS solutions are deployed in Layer 3 (network layer).



- With the rapid emergence of Web applications, the demand for application-level QoS control on local computers is growing.
- The disorderly competition in bandwidth usage may cause bandwidth-sensitive applications working abnormally.
- Although the traditional fixed bandwidth allocation strategy can avoid this problem and guarantee the QoS, it can lead to bandwidth idling.

- Various compression techniques and a host of coding techniques to adapt application traffic to network conditions and to economize the use of bandwidth.
  - HTML, MPEG, MP3 etc.
- Various application protocols and standards to meet specific needs of application users.
  - Internet Explorer, Chrome Navigator, etc. are used for web browsing.
  - Microsoft Outlook Express, WhatsApp, Zoom , etc. are used for e-mail and chat.
  - developing playback buffer to reduce the effect of network fluctuations such as jitter.

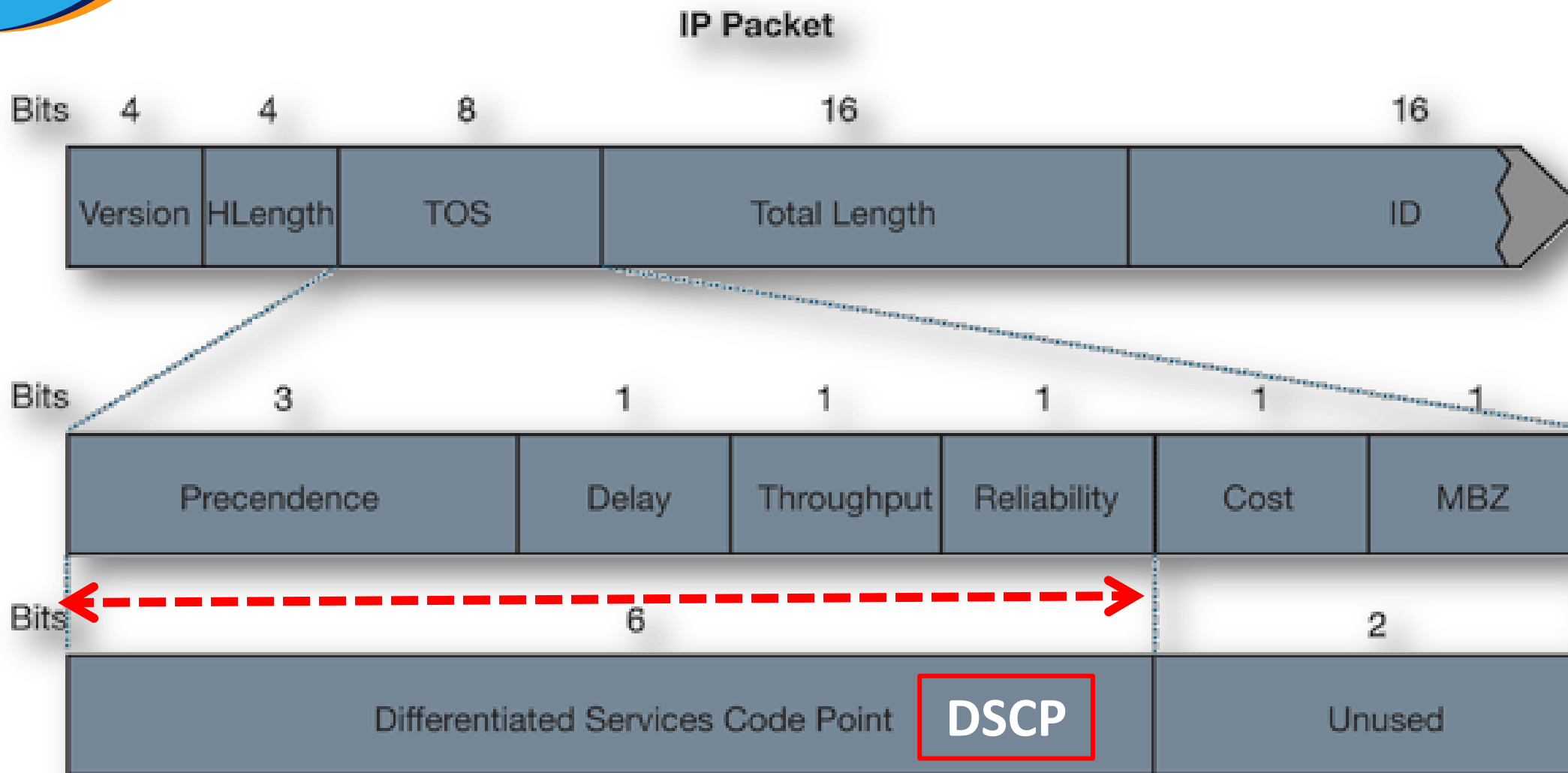


- TCP and UDP use a checksum facility in the header field of the datagram they transport to check the integrity of the datagram.
- TCP is with built-in traffic management and control mechanisms.
- TCP has both a packet rate control and a congestion control mechanism that will reduce packet loss and build-up of packet delay.
- Each correctly received TCP packet is acknowledged thereby providing guaranteed service to the sender.
- TCP QoS execution is suitable for old applications.
- TCP cannot provide a consistent level of service that will satisfy the need of real time applications.

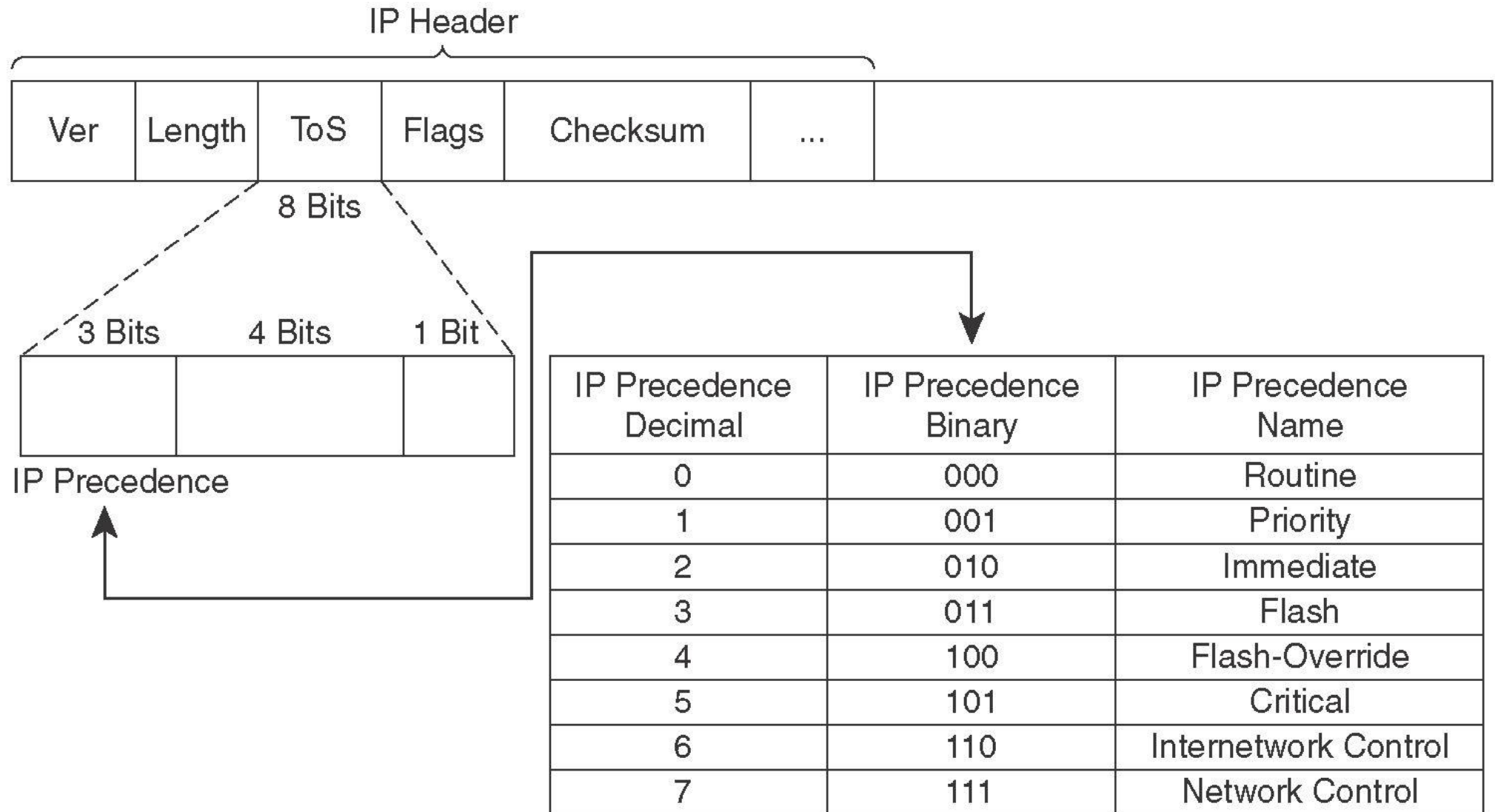
- 1) IP protocol does not provide guaranteed service.
- 2) IP has implicit QoS provision.
- 3) The Type of Service (ToS) byte field in IPv4 packet header, which has so far not been used, was designed for QoS control.
- 4) A checksum facility is also available in the packet header to check if the packet header has been corrupted.

- 5) IP can use a protocol, such as Internet Control Message Protocol (ICMP) to communicate error messages such as packet loss.
- 6) It also could be used to transmit or relay various control messages and configuration information from receiver host to sender host.
- 7) These functions, add together to enhance QoS for packet delivery in the network layer.

# Type of service (ToS)



## ToS (IP Precedence)



The IP Type of Service has the following fields:

Bits 0-2: Precedence.

Bit 3: 0 = Normal Delay, 1 = Low Delay.

Bits 4: 0 = Normal Throughput, 1 = High Throughput.

Bits 5: 0 = Normal Reliability, 1 = High Reliability.

Bit 6-7: Reserved for Future Use.

(CU: currently unused)

| 0          | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|------------|---|---|---|---|---|---|---|
| PRECEDENCE |   |   | D | T | R | 0 | 0 |

111 - Network Control

110 - Internetwork Control

101 - CRITIC/ECP

100 - Flash Override

011 - Flash

010 - Immediate

001 - Priority

000 - Routine



# DSCP to Service Class Mapping

| Service Class Name      | DSCP Name         | DSCP Value              | Application Examples                             |
|-------------------------|-------------------|-------------------------|--------------------------------------------------|
| Network Control         | CS6               | 110000                  | Network routing                                  |
| Telephony               | EF                | 101110                  | IP Telephony bearer                              |
| Signaling               | CS5               | 101000                  | IP Telephony signaling                           |
| Multimedia Conferencing | AF41,AF42<br>AF43 | 100010,100100<br>100110 | H.323/V2 video conferencing (adaptive)           |
| Real-Time Interactive   | CS4               | 100000                  | Video conferencing and Interactive gaming        |
| Multimedia Streaming    | AF31,AF32<br>AF33 | 011010,011100<br>011110 | Streaming video and audio on demand              |
| Broadcast Video         | CS3               | 011000                  | Broadcast TV & live events                       |
| Low-Latency Data        | AF21,AF22<br>AF23 | 010010,010100<br>010110 | Client/server transactions<br>Web-based ordering |
| OAM                     | CS2               | 010000                  | OAM&P                                            |
| High-Throughput Data    | AF11,AF12<br>AF13 | 001010,001100<br>001110 | Store and forward applications                   |
| Standard                | DF (CS0)          | 000000                  | Undifferentiated applications                    |
| Low-Priority Data       | CS1               | 001000                  | Any flow that has no BW assurance                |

CS No.

CS = 000

<https://www.tucny.com/Home/dscp-tos>

- To Find the decimal value of an AF = XY  $\rightarrow 8x + 2y$   
 ex: AF32 =  $8 \times 3 + 2 \times 2 = 28$
- There are 4 different classes and each class will be placed in a different queue,
- Each class there is also a drop probability. When the queue is full, packets with a "high drop" probability will be deleted from the queue before the other packets.
- In total there are 3 levels for drop precedence. Low Medium High

The AF DSCP Values

|         | Drop Probability  |                   |                   |
|---------|-------------------|-------------------|-------------------|
| Class   | Low Drop          | Medium Drop       | High Drop         |
| Class 1 | AF11              | AF12              | AF13              |
|         | DSCP 10: (001010) | DSCP 12: (001100) | DSCP 14: (001110) |
| Class 2 | AF21              | AF22              | AF23              |
|         | DSCP 18: (010010) | DSCP 20: (010100) | DSCP 22: (010110) |
| Class 3 | AF31              | AF32              | AF33              |
|         | DSCP 26: (011010) | DSCP 28: (011100) | DSCP 30: (011110) |
| Class 4 | AF41              | AF42              | AF43              |
|         | DSCP 34: (100010) | DSCP 36: (100100) | DSCP 38: (100110) |

AF (Class No. + No. + 0)

- The type of service (ToS) field in the IPv4 packet header is mapped identically to the Traffic Class field in IPv6, and it is used in the same fashion. With IPv6, however, several additional classifiers must be considered, all related to the IPv6 packet header format.
- DSCP (Differentiated Services Code Point) marking uses 6-bits of the 8-bit ToS field in the IP Header to provide up to 64 classes (or code points) for traffic.
- Since DSCP is a layer 3 marking method, there is no concern about compatibility as there is with 802.1p marking. Devices that do not support DSCP will simply ignore the tags, or at worst, they will reset the tag value to 0.

<https://what-when-how.com/ipv6-for-enterprise-networks/quality-of-service-qos-ipv6/>

[https://help.sonicwall.com/help/sw/eng/5800/25/9/0/content/Ch78\\_Firewall\\_Managing\\_QoS.088.08.html](https://help.sonicwall.com/help/sw/eng/5800/25/9/0/content/Ch78_Firewall_Managing_QoS.088.08.html)

## IPv4 Header

|                     |          |                 |              |                 |
|---------------------|----------|-----------------|--------------|-----------------|
| Version             | IHL      | Type of Service | Total Length |                 |
| Identification      |          |                 | Flags        | Fragment Offset |
| Time to Live        | Protocol | Header Checksum |              |                 |
| Source Address      |          |                 |              |                 |
| Destination Address |          |                 |              |                 |
| Options             |          |                 |              | Padding         |

## IPv6 Header

|                     |               |            |             |           |
|---------------------|---------------|------------|-------------|-----------|
| Version             | Traffic Class | Flow Label |             |           |
| Payload Length      |               |            | Next Header | Hop Limit |
| Source Address      |               |            |             |           |
| Destination Address |               |            |             |           |

### Legend

- Field's Name Kept from IPv4 to IPv6
- Fields Not Kept in IPv6
- Name and Position Changed in IPv6
- New Field in IPv6



### IPv6 QoS

#### RFC 2460/3697

Currently IPv6 provides support for QoS marking via a field in the IPv6 header.

Similar to the type of service (ToS) field in the IPv4 header, the traffic class field (8 bits) is available for use by originating nodes and/or forwarding routers to identify and distinguish between different classes or priorities of IPv6 packets.

**Figure 1**

The traffic class field may be used to set specific precedence or differentiated services code point (DSCP) values. These values are used in the exact same way as in IPv4.

The key advantage with the flow label is that the transit routers do not have to open the inner packet to identify the flow, which aids with identification of the flow when using encryption and other scenarios.



Current Cisco IOS® Software support for IPv6 QoS includes:

- Packet classification
- Queuing (includes LLQ; excludes legacy PQ/CQ)
- Traffic shaping
- WRED

IPv6 also has a 20-bit field known as the flow label field (RFC 3697). The flow label enables per-flow processing for differentiation at the IP layer.

It can be used for special sender requests and is set by the source node.

The flow label must not be modified by an intermediate node.

Planned Cisco IOS Software support for IPv6 QoS includes:

- Compressed Real-Time Protocol (cRTP)
- Network-based application recognition (NBAR)
- Committed access rate (CAR)

- **Default PHB**—With the 3 most significant bits of the DiffServ/DSCP field set to 000, the Default PHB is used for best effort (BE) service. If the DSCP value of a packet is not mapped to a PHB, it is consequently assigned to the default PHB.
- **Assured forwarding (AF) PHB**—With the most significant 3 bits of the DSCP field set to 001, 010, 011, or 100 (these are also called AF1, AF2, AF3, and AF4), the AF PHB is used for guaranteed bandwidth service.
- **Expedited forwarding (EF) PHB**—With the most significant 3 bits of the DSCP field set to 101 (the whole DSCP field is set to 101110, decimal value of 46), the EF PHB provides low delay service.

<http://www.patrickdenis.biz/blog/ip-precedence-dscp-tos-lookup-table/>



- **There is 3 different overall Models for QoS :**
  - 1) Best Effort is when no QoS is apply to packets , FIFO for high speed interfaces and scalable with no Overhead.
  - 2) IntServ (Integrated Services) is where devices/applications exchanges reservations information to enforce QoS ( not scalable )
    - the main process is Resource Reservation Protocol ( RSVP ) it use PATH and RESV messages. Some called this Hard QoS.
    - Resource Reservation — Device reserve appropriate bandwidth for their needed flow.
    - Admission Control — Device decide when reservation request should be rejected.
  - 3) DiffServ (Differentiated Services) is where devices don't have to exchange any reservation information but you have to configure them separately.

There is a quite easy way to get from one to the other:

- 1)  $DSCP * 4 = ToS$ ,
- 2) or you can use following table  
(<https://www.tucny.com/Home/dscp-tos>).

```
C:\WINDOWS\system32\cmd.exe

Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
          [-r count] [-s count] [[-j host-list] ! [-k host-list]]
          [-w timeout] [-R] [-S srcaddr] [-c compartment] [-p]
          [-4] [-6] target_name

Options:
    -t          Ping the specified host until stopped.
                To see statistics and continue - type Control-Break;
                To stop - type Control-C.
    -a          Resolve addresses to hostnames.
    -n count    Number of echo requests to send.
    -l size     Send buffer size.
    -f          Set Don't Fragment flag in packet (IPv4-only).
    -i TTL      Time To Live.
    -v TOS      Type Of Service (IPv4-only. This setting has been deprecated
                and has no effect on the type of service field in the IP
                Header).
    -r count    Record route for count hops (IPv4-only).
    -s count    Timestamp for count hops (IPv4-only).
    -j host-list Loose source route along host-list (IPv4-only).
    -k host-list Strict source route along host-list (IPv4-only).
    -w timeout  Timeout in milliseconds to wait for each reply.
    -R          Use routing header to test reverse route also (IPv6-only).
                Per RFC 5095 the use of this routing header has been
                deprecated. Some systems may drop echo requests if
                this header is used.
    -S srcaddr  Source address to use.
    -c compartment Routing compartment identifier.
    -p          Ping a Hyper-V Network Virtualization provider address.
    -4          Force using IPv4.
    -6          Force using IPv6.

C:\Users\user>ping -v 184 www.google.com

Pinging www.google.com [142.250.200.196] with 32 bytes of data:
Reply from 142.250.200.196: bytes=32 time=66ms TTL=117
Reply from 142.250.200.196: bytes=32 time=59ms TTL=117
Reply from 142.250.200.196: bytes=32 time=59ms TTL=117
Reply from 142.250.200.196: bytes=32 time=63ms TTL=117

Ping statistics for 142.250.200.196:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 59ms, Maximum = 66ms, Average = 61ms

C:\Users\user>
```

# Module 3

cmd → ping -v

**\*Wi-Fi**

Help Tools Wireless Telephony Statistics Analyze Capture Go View Edit File

Apply a display filter ... <Ctrl-/>

| DSCP             | Info                                                                  | Length | Protocol | Destination                    | Source            | Time      | .No |
|------------------|-----------------------------------------------------------------------|--------|----------|--------------------------------|-------------------|-----------|-----|
|                  | Who has 192.168.1.19? Tell 192.168.1.1                                | 42     | ARP      | LiteonTe_99:ec:33              | HuaweiTe_41:12:3c | 10.816408 | 11  |
|                  | is at 28:e3:47:99:ec:33 192.168.1.19                                  | 42     | ARP      | HuaweiTe_41:12:3c              | LiteonTe_99:ec:33 | 10.816438 | 12  |
|                  | Who has 192.168.1.1? Tell 192.168.1.9                                 | 42     | ARP      | Broadcast                      | SamsungE_0d:ff:f6 | 11.507311 | 16  |
|                  | Standard query 0xf36d SRV _VLMCS._TCP.home                            | 96     | DNS      | fe80::1 ...fe80::5110:3ec0:313 | 21.326725         | 31        |     |
|                  | Standard query 0xf36d SRV _VLMCS._TCP.home                            | 96     | DNS      | fe80::1 ...fe80::5110:3ec0:313 | 24.338755         | 33        |     |
|                  | Neighbor Solicitation for fe80::1 from 28:e3:47:99:ec:33              | 86     | ICMPv6   | fe80::1 ...fe80::5110:3ec0:313 | 26.322329         | 38        |     |
|                  | Neighbor Advertisement fe80::1 (rtr, sol)                             | 78     | ICMPv6   | ...fe80::5110:3ec0:313         | fe80::1           | 26.324918 | 39  |
|                  | Standard query 0xd688 A updatekeepalive.mcafee.com                    | 106    | DNS      | fe80::1 ...fe80::5110:3ec0:313 | 26.935403         | 40        |     |
|                  | Standard query response 0xd688 A updatekeepalive.mcafee.com CNAME upd | 158    | DNS      | ...fe80::5110:3ec0:313         | fe80::1           | 26.961559 | 41  |
| Class Selector 3 | Echo (ping) reply id=0x0001, seq=5/1280, ttl=117 (request in 8)       | 74     | ICMP     | 192.168.1.19                   | 142.250.200.196   | 10.095725 | 4   |
| Class Selector 3 | Echo (ping) reply id=0x0001, seq=6/1536, ttl=117 (request in 14)      | 74     | ICMP     | 192.168.1.19                   | 142.250.200.196   | 11.114417 | 15  |
| Class Selector 3 | Echo (ping) reply id=0x0001, seq=7/1792, ttl=117 (request in 19)      | 74     | ICMP     | 192.168.1.19                   | 142.250.200.196   | 12.133358 | 20  |

C:\WINDOWS\system32\cmd.exe

**for DSCP 26**

```
C:\Users\user>ping -v 104 www.google.com

Pinging www.google.com [142.250.200.196] with 32 bytes of data:
Reply from 142.250.200.196: bytes=32 time=64ms TTL=117
Reply from 142.250.200.196: bytes=32 time=61ms TTL=117
Reply from 142.250.200.196: bytes=32 time=60ms TTL=117
Reply from 142.250.200.196: bytes=32 time=66ms TTL=117

Ping statistics for 142.250.200.196:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 60ms, Maximum = 66ms, Average = 62ms

C:\Users\user>
```

**Wireshark · Packet 9 · Wi-Fi**

74 bytes captured (592 bits) on interface \Device\NPF\_{16E21364-2D10-4D9E-AC61-0F560444B8F8}, id 0

Ethernet II, Src: HuaweiTe\_41:12:3c (b4:f5:8e:41:12:3c), Dst: LiteonTe\_99:ec:33 (28:e3:47:99:ec:33)

Destination: LiteonTe\_99:ec:33 (28:e3:47:99:ec:33)

Source: HuaweiTe\_41:12:3c (b4:f5:8e:41:12:3c)

Type: IPv4 (0x0800)

Internet Protocol Version 4, Src: 142.250.200.196, Dst: 192.168.1.19

Version: 4 = .... 0100

Header Length: 20 bytes (5) = 0101 ....

**Differentiated Services Field: 0x60 (DSCP: CS3, ECN: Not-ECT)**

Differentiated Services Codepoint: Class Selector 3 (24) = ..00 0110

Explicit Congestion Notification: Not ECN-Capable Transport (0) = 00.. ....

Total Length: 60

Identification: 0x0000 (0)

0000 28 e3 47 99 ec 33 b4 f5 8e 41 12 3c 08 00 45 60 (.G..3..A.<...E`

0010 00 3c 00 00 00 00 75 01 2b e7 8e fa c8 c4 c0 a8 <.....u. +.....

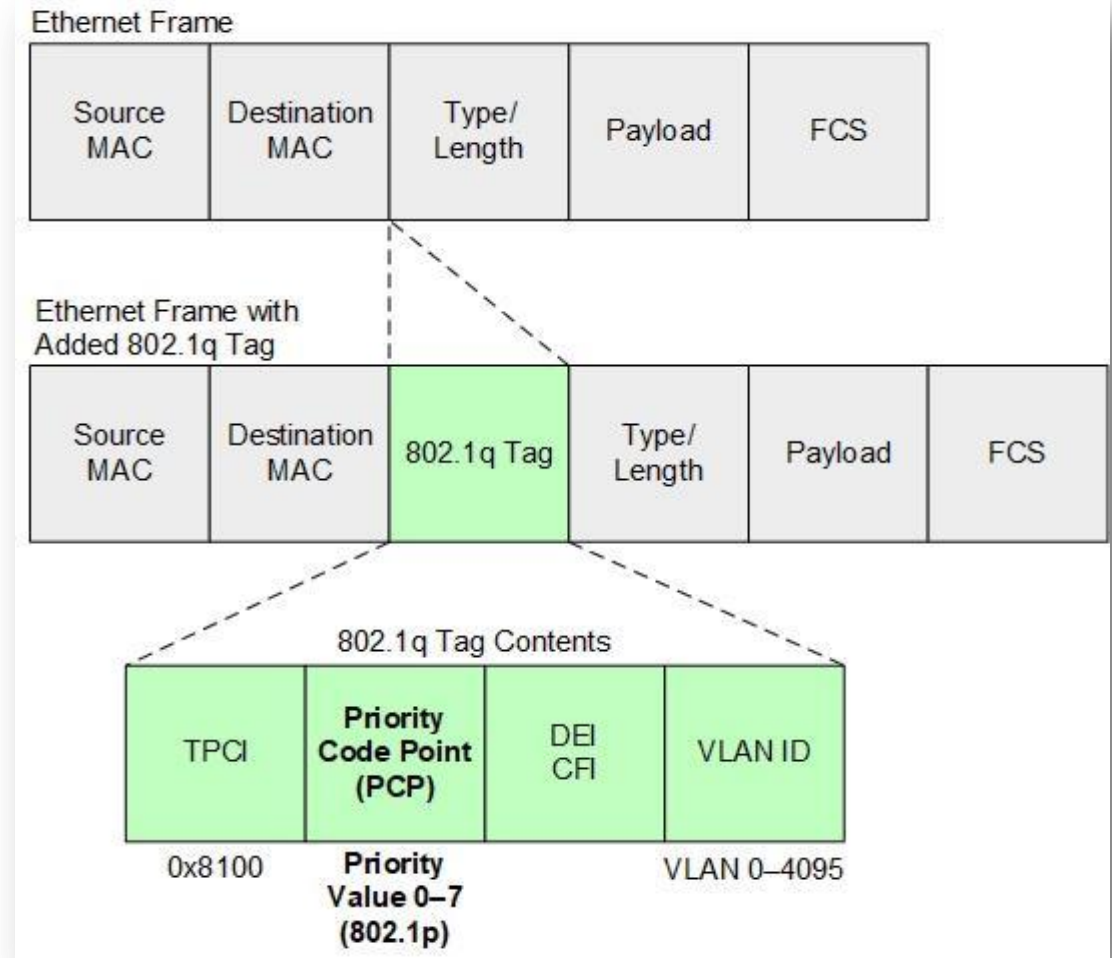
0020 01 13 00 00 55 56 00 01 00 05 61 62 63 64 65 66 ....UV.. ..abcdef

0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghijklmn opqrstuv

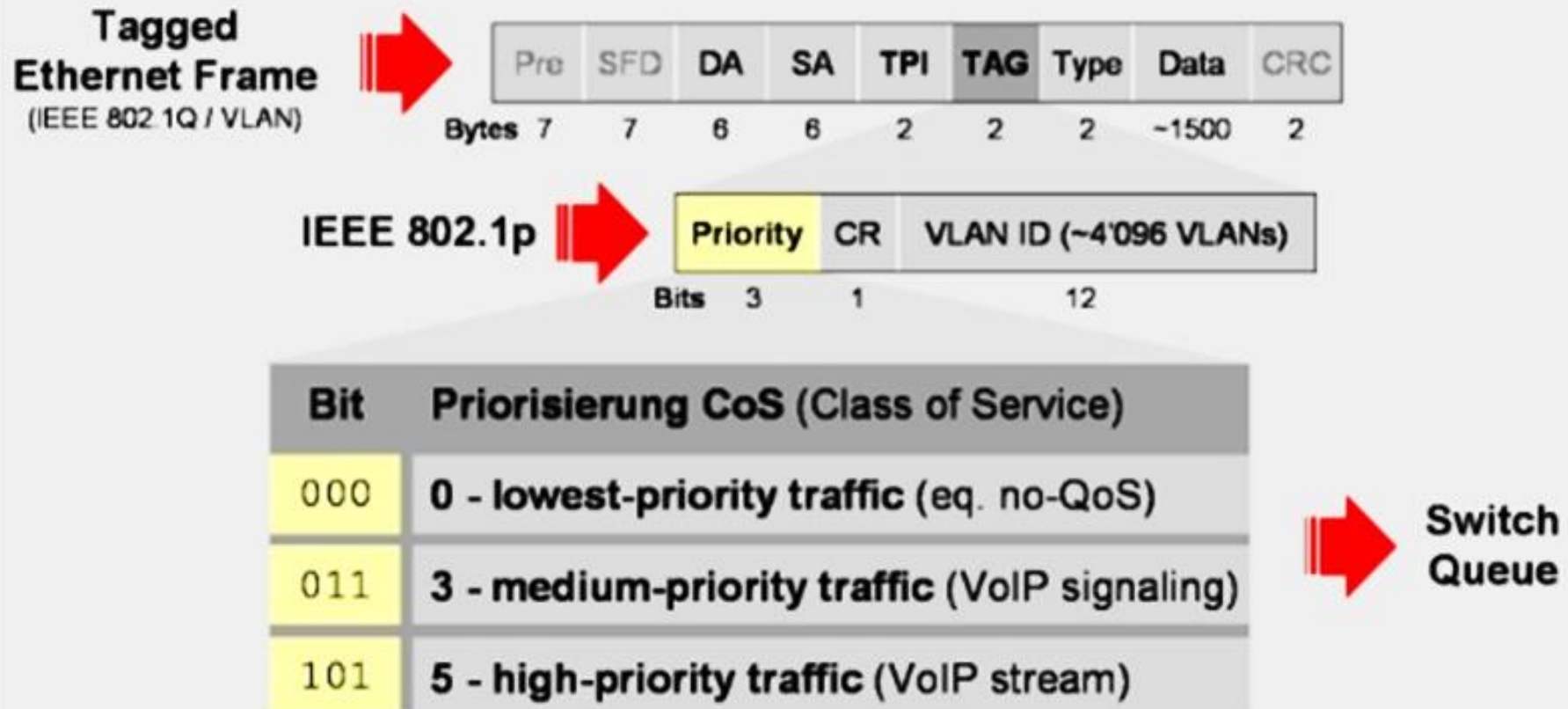
0040 77 61 62 63 64 65 66 67 68 69 wabcdefg hi

- The Internet link layer is made-up of diverse network technologies with disparate QoS offerings.
- Although they generally provide the same best-effort service to datagrams from higher-layer protocols such as Internet Protocol (IP), each of them traditionally has its own style of QoS built into it.
- For example: Ethernet or Token Ring has a contention resolution algorithm built into its medium access mechanism to limit packet loss as a result of traffic collision. The Ethernet Carrier Sense Multiple Access with Collision Detention (CSMA/CD) is a highly efficient and popular media access mechanism. → (CSMA/CA in adHoc)

- One of the layer 2 method is the IEEE 802.1p standard, wherein 3-bits of an additional 16-bits inserted into the header of the Ethernet frame can be used to designate the priority of the frame.



## QoS/CoS (VLAN Layer-2 Priorisierung)



- The **8021p-inbound** command maps the 802.1p priority of incoming VLAN packets in a DiffServ domain to the PHB and colors the packets (  **HUAWEI** ).

| Parameter          | Description                                    | Value                                                                                        |
|--------------------|------------------------------------------------|----------------------------------------------------------------------------------------------|
| <i>8021p-value</i> | Specifies the 802.1p priority of VLAN packets. | The value is an integer that ranges from 0 to 7. A larger value indicates a higher priority. |

Mappings from 802.1p priorities to PHBs

| 802.1p Priority | PHB |
|-----------------|-----|
| 0               | BE  |
| 1               | AF1 |
| 2               | AF2 |
| 3               | AF3 |
| 4               | AF4 |
| 5               | EF  |
| 6               | CS6 |
| 7               | CS7 |



How to setup QoS in your modem?

<https://www.tp-link.com/ae/support/faq/535/>

- **pip install NetfilterQueue**

- 1) NetfilterQueue provides access to packets matched by an iptables rule in **Linux**. Packets so matched can be accepted, dropped, altered, reordered, or given a mark.
- 2) The current version of NetfilterQueue requires Python 3.6 or later. The last version with support for Python 2.7 was 0.9.0.

- **Set Socket Options → setsockopt()**
  - For IPv4, you need to use IPPROTO\_IP/IP\_TOS.
  - For IPv6 you need to use IPPROTO\_IPV6/IPV6\_TCLASS.
    - In the DSCP value: bits shifted to the left 2 places, since DSCP is a 6 bit field, but IP\_TOS and IPV6\_TCLASS are 8 bit fields.
  - IPV6\_RECVTCLASS (set, get → Boolean/DWORD)
  - SOL\_SOCKET/SO\_PRIORITY
  - ICMP → traffic\_class

### level

(Input) Whether the request applies to the socket itself or the underlying protocol being used. Supported values are:

|                       |                                           |
|-----------------------|-------------------------------------------|
| <i>IPPROTO_IP</i>     | Request applies to IP protocol layer.     |
| <i>IPPROTO_TCP</i>    | Request applies to TCP protocol layer.    |
| <i>SOL_SOCKET</i>     | Request applies to socket layer.          |
| <i>IPPROTO_IPV6</i>   | Request applies to IPv6 protocol layer.   |
| <i>IPPROTO_ICMPV6</i> | Request applies to ICMPv6 protocol layer. |

```
s = socket.socket(socket.AF_INET, socket.SOCK_RAW, socket.IPPROTO_IP)
```

- **IPPROTO\_IP** creates a socket that sends/receives raw data for IPv4-based protocols (TCP, UDP, etc). It will handle the IP headers for you, but you are responsible for processing/creating additional protocol data inside the IP payload.

[https://www.ibm.com/docs/ja/i/7.2?topic=ssw\\_ibm\\_i\\_72/apis/ssocko.htm](https://www.ibm.com/docs/ja/i/7.2?topic=ssw_ibm_i_72/apis/ssocko.htm)

- Set or receive the Type-Of-Service (TOS) field.
- There are some standard ToS flags defined:
  - IPTOS\_LOWDELAY to minimize delays for interactive traffic.
  - IPTOS\_THROUGHPUT to optimize throughput.
  - IPTOS\_RELIABILITY to optimize for reliability.
  - IPTOS\_MINCOST should be used for "filler data" where slow transmission doesn't matter.
- Some high priority levels may require super user privileges (ADMIN ).
- The priority can also be set in a protocol independent way by the (SOL\_SOCKET, SO\_PRIORITY) socket option.

```
import socket

# the public network interface
HOST = socket.gethostname(socket.gethostname())
print('HOST=', HOST)
# create a raw socket and bind it to the public interface
s = socket.socket(socket.AF_INET, socket.SOCK_RAW, socket.IPPROTO_IP)
s.bind((HOST, 0))

# Include IP headers
s.setsockopt(socket.IPPROTO_IP, socket.IP_HDRINCL, 1)
s.setsockopt(socket.IPPROTO_IP, socket.IP_TOS, 1)

print('IP_HDRINCL = ', s.getsockopt(socket.IPPROTO_IP, socket.IP_HDRINCL))
print('IP_TOS = ', s.getsockopt(socket.IPPROTO_IP, socket.IP_TOS))

# receive all packages
s.ioctl(socket.SIO_RCVALL, socket.RCVALL_ON)

# receive a package
print(s.recvfrom(65565))
print(s)
```

- RFC 3168 [Ramakrishnan, Floyd, and Black 2001] contains the definition of the ECN field (Part of ToS). Applications should generally leave the setting of the ECN field to the kernel, and should specify zero values in the low two bits of the value set with **IP\_TOS**.
- **QoS API**: is the only interface that programmers will require to create QoS-aware or QoS-enabled applications. Most operations that happen on behalf of an application in the QoS sequence are a result of QoS API calls communicating requests down (and sometimes back up) through QoS components.

[https://www.masterraghu.com/subjects/np/introduction/unix\\_network\\_programming\\_v1.3/ch07lev1sec6.html](https://www.masterraghu.com/subjects/np/introduction/unix_network_programming_v1.3/ch07lev1sec6.html)

- **Implementing QoS in Windows Environments**

<https://docs.microsoft.com/en-us/windows-server/networking/technologies/qos/qos-policy-top>

<https://support.talkdesk.com/hc/en-us/articles/360025362612-Implementing-QoS-in-Windows-Environments>

- Indicates whether the IP stack should populate the control buffer with a message containing the Traffic Class IPv6 header field on a received datagram.
- When this value is true, the [WSARecvMsg](#) function will return optional control data containing the Traffic Class IPv6 header field value of the received datagram.
- This option allows the Traffic Class IPv6 header field of the received datagram to be returned in the [WSAMSG](#) structure. The returned message type will be IPV6\_TCLASS.
- All DSCP bits of the Traffic Class field will be returned.
- This option is only valid on datagram sockets (the socket type must be SOCK\_DGRAM).

- `traffic_class`

The traffic class of ICMP packets. Provides a defined level of service to packets by setting the DS Field (formerly TOS) or the Traffic Class field of IP headers. Packets are delivered with the minimum priority by default (Best-effort delivery). Intermediate routers must be able to support this feature.

*Only available on Unix systems. Ignored on Windows.*