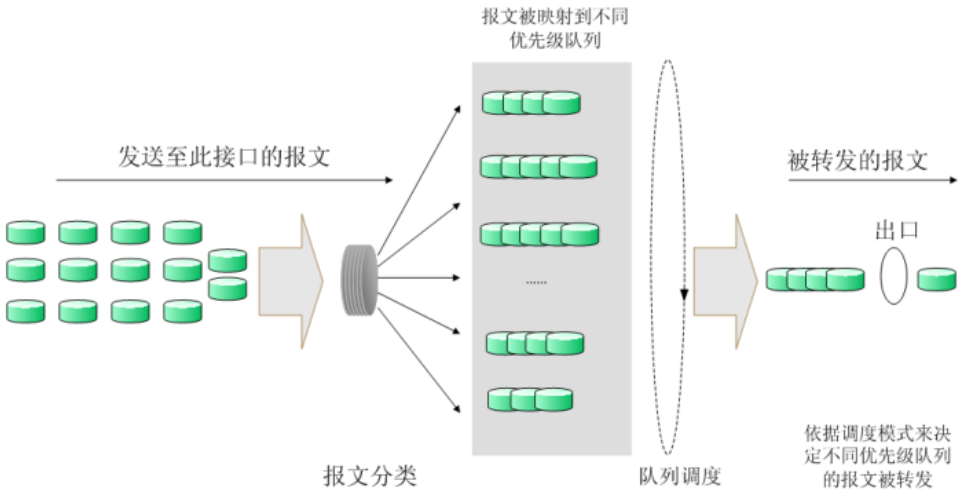
The so-called QoS is for different needs of various network applications, and provides different quality of service, such as providing dedicated bandwidth, reducing packet loss rate, reducing packet transmission delay and delay jitter. That is to say, in the case that the bandwidth is not sufficient, the contradiction between the bandwidth occupied by various service flows is balanced.

The switch classifies the data streams in the ingress phase, and then maps different types of data streams to different priority queues in the export phase. Finally, the scheduling mode determines the manner in which packets of different priority queues are forwarded, thereby implementing QoS features.



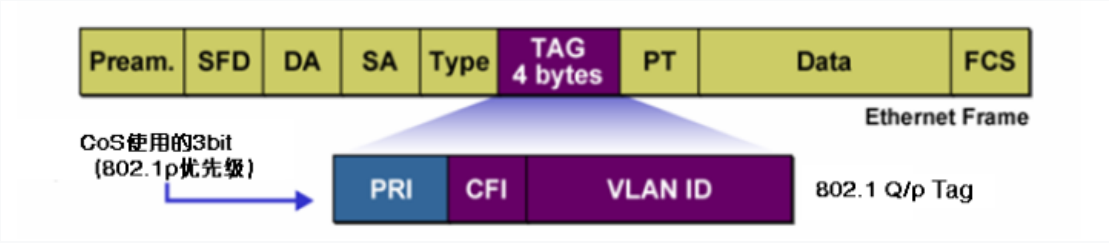
QoS working principle

1.Message classification: objects are identified according to certain matching rules.
2.Mapping: Users can map packets entering the switch to different priority queues according to the priority mode. The switch provides two priority modes: 802.1P priority and DSCP priority.
3.Queue scheduling: When the network is congested, it must solve the problem that multiple data streams compete for resources at the same time, usually by queue scheduling. The company's switches provide a total of five scheduling modes (supporting several of them depending on the switch chip), respectively

SP(Strict Priority)， RR(Round-Robin） WRR(Weighted Round-Robin） DRR(Deficit Round-Robin),WFQ(Weighted Fair Queuing）.

1、Basic concept of QoS queue scheduling

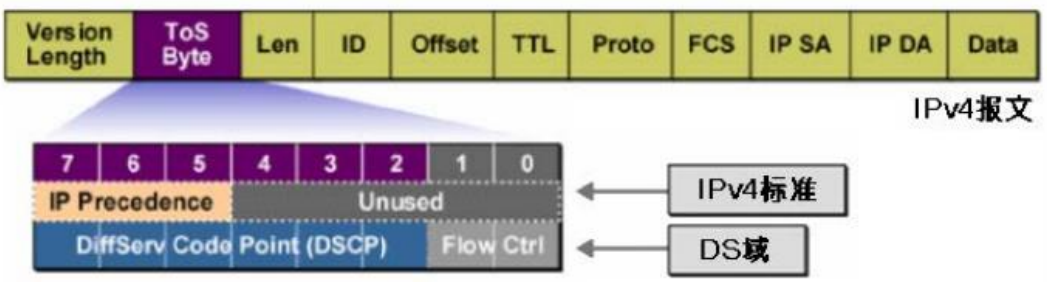
（1）802.1P priority



802.1Q Frame format

As shown in the figure, each 802.1Q Tag has a Pri field, which consists of three bits, ranging from 0 to 7. The 802.1P priority is based on the Pri field value to determine the priority of the data frame level. The Pri region can be configured with different priorities based on the configuration page of the switch. When the switch sends a data frame, it determines the priority of the transmission based on the tag of the data frame. For Untagged frames, the switch performs QoS processing on the data frames according to the default priority of the ingress port

（2）DSCP priority



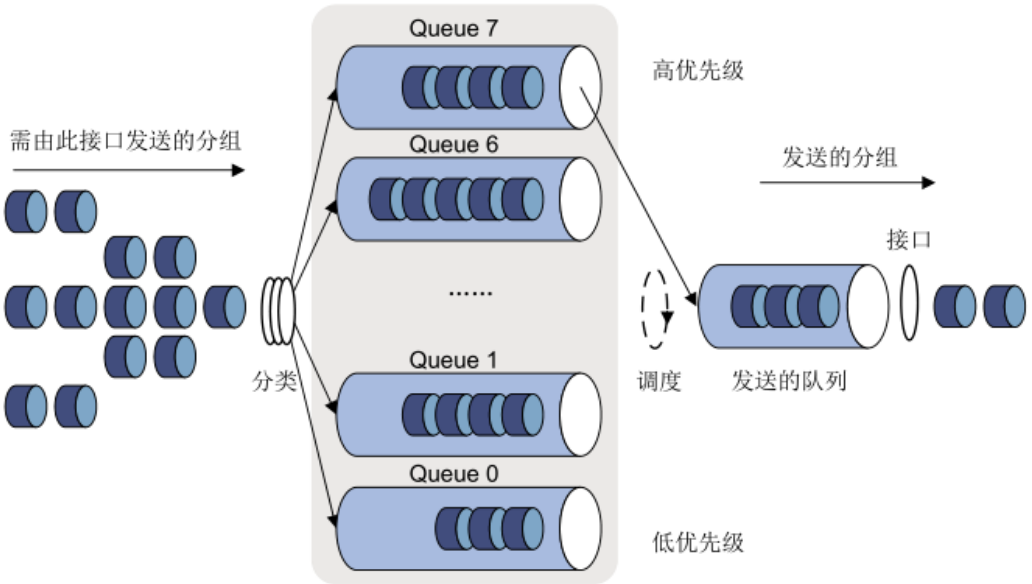
IP packet format

As shown in the figure, the ToS (Type of Service) field in the IP packet header has 8 bits. The first 3 bits represent the IP precedence, which ranges from 0 to 7. RFC 2474 redefines the ToS field of the IP packet header, which is called the DS domain. The DSCP (Differentiated Services Codepoint) priority is represented by the first 6 bits (0 to 5 bits) of the field, ranging from 0 to 63. The last 2 bits (6, 7 bits) are reserved bits. Through the configuration page of the switch, you can configure different DS fields to have different priorities. When the switch sends an IP packet, the switch determines the priority according to the DS domain of the IP packet. For non-IP packets, the switch decides which priority mode to use based on whether 802.1P priority is enabled and whether the data frame has a Tag.

2、QoS queue scheduling algorithm

When the network is congested, queue scheduling is usually used to solve the problem that multiple data streams compete for resources at the same time. The company's switches implement 8 or 4 scheduling queues depending on the switch chip. Queue 0 corresponds to the lowest priority queue, and Queue 7 corresponds to the highest priority queue.

（1） SP，Strict Priority：



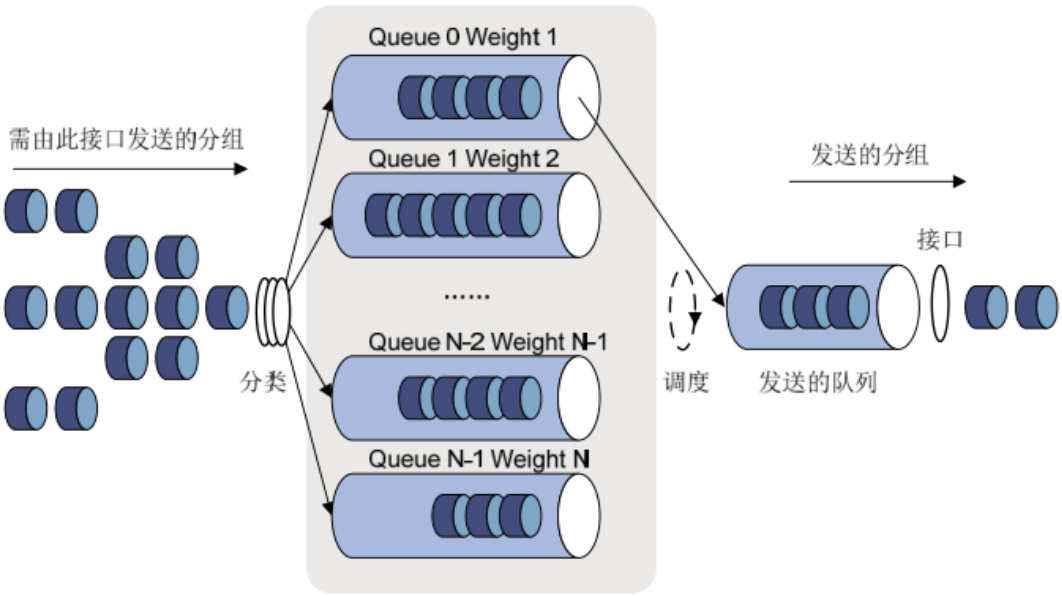
The scheduling mode of the SP mode is that the switch preferentially forwards the data frame with the highest priority at the current priority. After all the highest priority data frames are forwarded, the data frames of the next highest priority are forwarded. The switch has eight egress queues, which in turn are Queue 0-Queue 7. In SP queue mode, their priority increases in turn, and Queue 7 has the highest priority. The disadvantage of the SP queue is that if there is a packet in the higher priority queue for a long time when congestion occurs, the packet in the low priority queue will "starve" due to lack of service.

（2） RR，Round-Robin

The RR mode scheduling algorithm is that the circular queue avoids local queue starvation through the round robin service. The scheduler always moves sequentially to the next queue that has packets to send (empty queues are skipped). If each queue has packets waiting to be sent, the scheduling order matches the queue order; if some queues are empty, the other queues are frequently served. In extreme cases, if all other queues are empty, a single queue can use the full link bandwidth. When a packet enters an empty queue, the queue is serviced in the next cycle, thus avoiding queue "hungry."

The disadvantage of round-robin scheduling is that packet delays are difficult to improve, and it is not possible to allocate dedicated queues for low-latency traffic. The service interval of each queue depends entirely on how many packets are waiting to be sent in other queues during that time and the length of these packets. These variables are difficult to predict accurately, so RR scheduling is prone to delay jitter. The scheduler can schedule certain queues more frequently by changing the order of services (eg, using the order 1, 2, 3, 2, 4, 2, 1, 2, ...) to give these queues more frequent transmission opportunities, however packet size Random distribution still causes delay jitter problems.

（3） WRR，Weighted Round-Robin：



The WRR mode scheduling algorithm performs polling scheduling between queues according to the weight ratio to ensure that each queue gets a certain service time. The weighted value indicates the proportion of the acquired resource. The WRR queue avoids the disadvantage that packets in low priority may not be serviced for a long time when using SP scheduling, and although multiple queue scheduling is performed by polling, it is not a fixed allocation service time for each queue. If the queue is empty, the next queue schedule will be replaced immediately, so that the bandwidth resources can be fully utilized. The WRR algorithm is very similar to the DRR algorithm. The WRR algorithm uses a similar concept of time slice and difference, but the algorithm is slightly different. In WRR, the next queue is serviced when the number of bytes sent by the queue exceeds the limit allowed by the queue (still QN+DN). Therefore, the difference is a negative value (beyond the number of QN+DNs) and is taken as the reduction in the number of bytes sent by the queue in the next cycle.

（4） DRR，Deficit Round-Robin：

The DRR algorithm is an extension of the RR algorithm. The DRR algorithm assigns each queue a constant QN (time slice in weights) and a variable DN (difference). QN

reflects the long-term average number of bytes that the queue can send. The initial value of the DN is zero and is reset to 0 when the queue is empty. When the DRR algorithm serves a new queue, the scheduler resets the counter Bsent (indicating the number of bytes that the loop has sent from the queue). The DRR algorithm sends a packet from the queue when the following two conditions are met:
There are packets waiting to be sent in the queue;
 (QN+DN) is greater than or equal to (Bsent + the length of the next packet in the queue).
Otherwise, the difference DN+1 of the queue is set to QN+DN-Bsent, and the scheduler moves to the next queue in order. QN+DN indicates the maximum number of bytes that the queue can send during the service interval. To a certain extent, the DN can smooth the burst of the data stream. The queue can obtain long-term relative bandwidth allocation through QN. If the number of active queues is less than N, the active queue can share unused output link bandwidth based on the QN value.

（5） WFQ，Weighted Fair Queuing

WFQ is an abbreviation for Weighted Fair Queuing, which is a congestion algorithm that identifies conversations (in the form of data streams), separates packets belonging to individual conversations, and ensures that transmissions are fairly shared by these independent conversations. WFQ (weighted fair queuing weighted fair queuing) objectives: Provide a fair bandwidth allocation mechanism for each activity stream ,Provide faster scheduling mechanism for a small number of interactive streams ,Provide more bandwidth for high priority streams

WFQ: is a stream-based queuing algorithm. The arriving data stream is divided into multiple streams, and each stream is assigned to a FIFO queue. Dropping packets from the most active stream can provide faster service for inactive streams. WFQ is a fair-based queue based on Weight. The reason why WFQ is fair is that WFQ allocates the corresponding bandwidth according to the IP precedence of the packet. The packet with higher priority has more bandwidth and lower priority. Packets are allocated with less bandwidth, and all packets can be allocated bandwidth at any time, which is what makes it fair. When WFQ allocates bandwidth to packets based on IP precedence, it is allocated based on flows.

Comparison of WFQ algorithm with other scheduling algorithms:

In the DRR algorithm, each queue has a weight. The server polls each queue at a rate in a predetermined order. If an empty queue is encountered, the server immediately moves to the next queue. If the queue misses its transmission timing, it can only wait until the next timing belongs to it. If each queue is in use, the packets for that queue will not be processed until all queues have been processed. WFQ is not affected by this and is more suitable for processing variable length packets than DRR.