Title

Configuring QoS Policies

Successfully pass the free certification exam at IW Academy and become an Infinet Certified Engineer.

To the certification exam

- Introduction
- Terminology
- Packet distribution scheme
- Quality indicators
- QoS methods
- The traffic prioritization mechanism
- Throughput limitation mechanism
- Additional materials

Introduction

The evolution of the data networks entails an increase in the volume of the transmitted traffic, which requires the usage of a quality of service policy. The implementation of the policy will allow the classification of the network traffic and the distribution of the network resources between different traffic classes.

Terminology

- **QoS** (Quality of Service) - technology that allows to classify a data stream and to prioritize each stream's transmission in accordance with its class.
- **QoS policy** - document describing the principles of traffic stream classification and the resource requirements for each class.
- **Traffic stream** - the data of one service transmitted between two nodes.
- **Service** - process running on end nodes. The data of a service is distinguished by a unique set of service field values within the network packet's structure. IP telephony, web, and video surveillance are examples of services.
- **Responsibility area** - a network segment whose effective operation lays in the responsibility of a certain subject. A subject can be either a specific person or an organization.
- **DS domain** (Differentiated Services domain) - a logical area having uniform traffic classification rules, defined by a QoS policy. Usually the DS domain coincides with the responsibility area.
- **CIR** - Committed Information Rate. The system must guarantee the resource allocation in compliance with the CIR of the service.
- **MIR** - Maximum Information Rate. In case that the CIR is ensured, the additional resources may be allocated to the services. The additional resources cannot exceed the MIR threshold and their allocation is not guaranteed.

Packet distribution scheme

In packet networks, the traffic is transmitted from the sending node to the receiving node through communication channels and intermediate devices. Generally a data packet is processed by each intermediate device independently. Let's look at an example of data packet processing performed by an intermediate network device (Figure 1):

1. Node-2 generates a data frame and transmits it to Medium-2. The data is encapsulated in a frame based on the L2-protocol that is used in Medium-2.
2. The data frame is distributed in Medium-2: the frame is converted into a modulated signal according to the physical properties of the environment. The signals used in wired and wireless environments are different and this affects their propagation properties and their usage scenarios.
3. The signal arrives at the incoming network interface of the intermediate network device; after demodulation, the received data frame is checked for integrity: the damaged frames are discarded.
4. Next, the frame must be processed by the switching module in order to determine its path. If the frame is addressed to this intermediate network device, it will be passed for processing to the internal services. If the frame is addressed to another node, two scenarios are possible: the frame is passed to further processing until it reaches the output interface, or it is discarded (if Medium-2 is a common environment, where all signals will be received by all devices connected to the medium, according to the L2 protocol's operational principles, if the destination address in the frame's header does not belong to the device, then the device should discard it).
5. In case that the frame should be processed and transferred to another node, before exiting the device it will be placed into a packet queue. A packet queue is a set of buffers that contain the data received by the incoming interfaces. The number and size of the memory buffers used for the packet storage are not standardized and depend on the equipment's manufacturer. For example, the InfiLINK 2x2 family of devices has 32 queues, 17 of which are available for configuration to the user.
6. The data frame passes through the packet queue to which it was assigned and arrives at the outgoing interface.
7. Since packet queues are a link between incoming and outgoing interfaces, a device should have a controller that fills the queues with the incoming data and picks data from the queues for transmission to the outgoing interfaces. Usually, these functions are performed by the central processing unit (CPU). As it will be shown below, the filling and the emptying of data into and from the queues can be performed unevenly and depends on the classification of the data streams.
8. The outgoing interface generates a modulated signal and transmits it to Medium-5 which is connected to Node-5, the destination of the original data frame.

9. Node-5 receives the signal, demodulates it and processes the received data frame.

Note that in modern network devices, the network interfaces are usually combined and can operate both as incoming and outgoing.

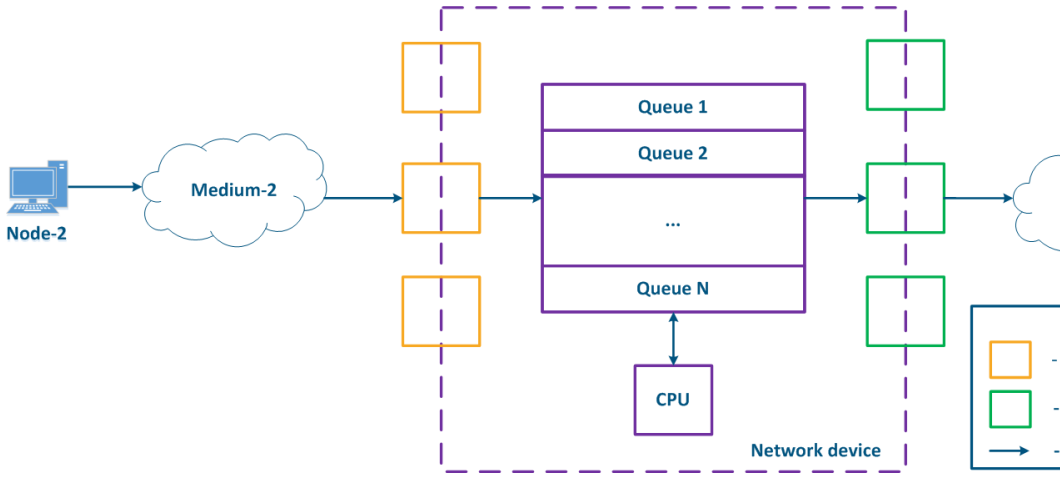


Figure 1 - Traffic passing through an intermediate network device

A network device can be intermediate for several pairs of nodes and each node can transmit the data of several services (Figure 2a). Let's look at a scheme where the "Network device" is an intermediate node for the traffic coming from the following pairs of nodes: Node-1 - Node-4, Node-2 - Node-5 and Node-3 - Node-6. The first pair transmits data for three services, the second for two and the third for one service. If there are no QoS settings, the data of all services get through the general queue in the order they are received at the "Network device" and in the same order they will be transferred from the queue to the outgoing interfaces.

With QoS configured, each of the incoming traffic flows can be classified by its type (for example) and a separate queue can be mapped to each class (Figure 2b). Each packet queue can be assigned a priority, which will be taken into account while extracting the packets from the queues, and will guarantee specific quality indicators. The traffic flow classification can be performed not only with respect to the services used, but according to other criteria also. For example, each pair of nodes can be assigned to a separate packet queue (Figure 2c).

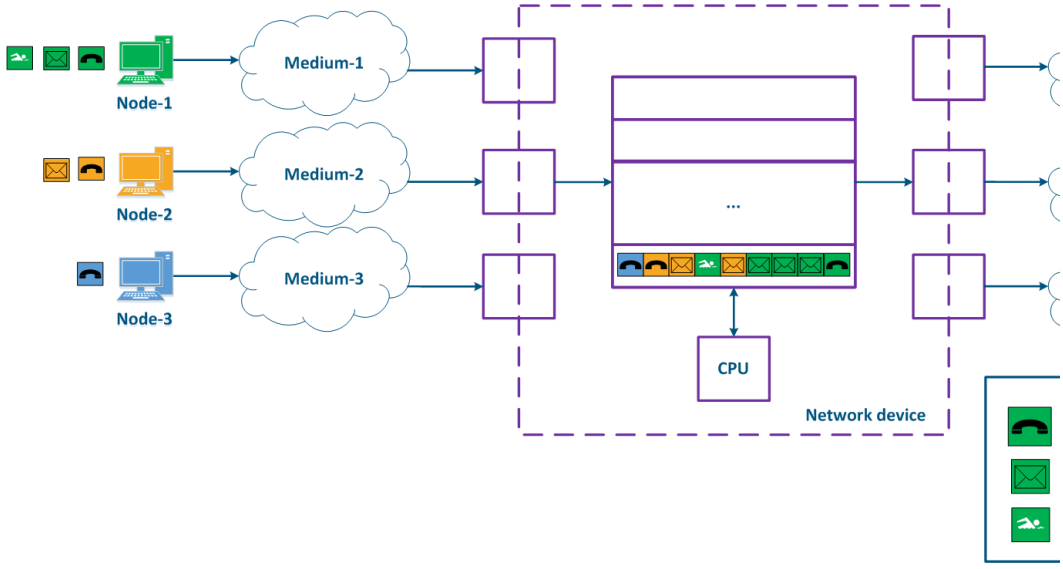


Figure 2a - Queuing for various services without QoS

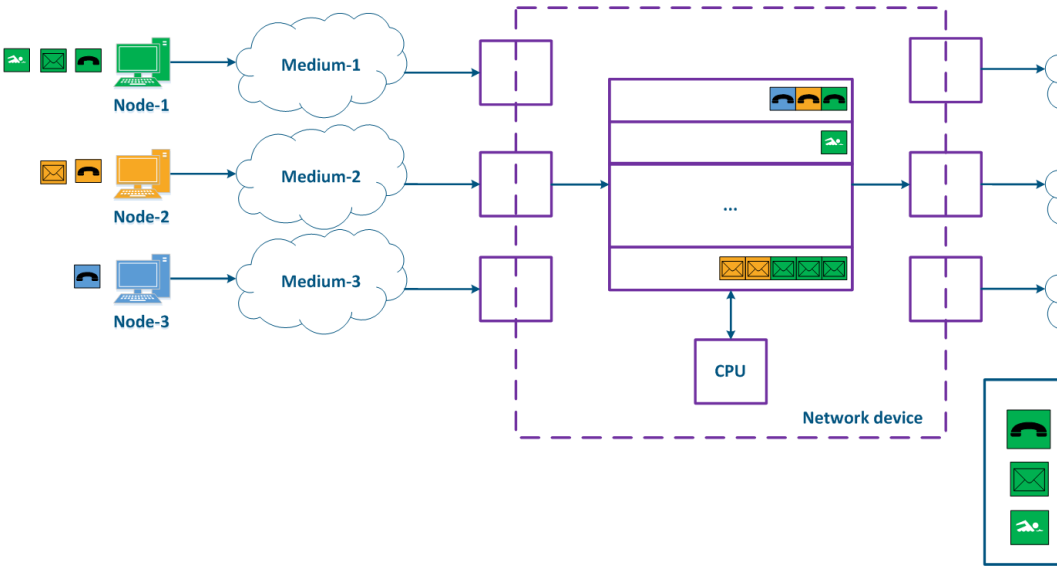


Figure 2b - Queuing for various services with QoS

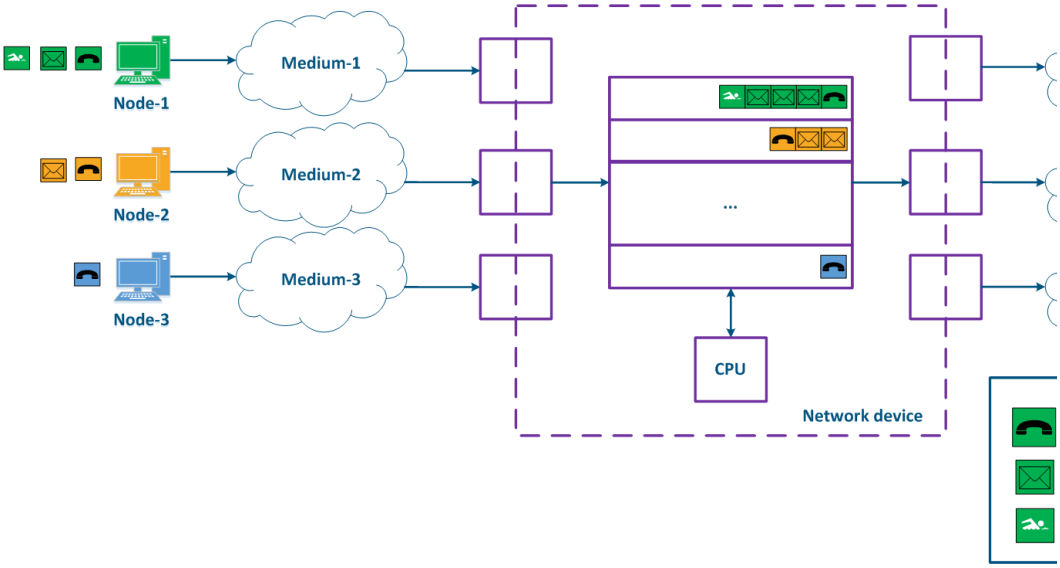


Figure 2b - Queuing for various users with QoS

Keep in mind that several intermediate network devices can be located on the data path between the source and the receiver, having independent packet queues, i.e. an effective QoS policy implementation will require the configuration of several network nodes.

Quality indicators

The main conclusions from the previous section, which will be used to define the quality metrics, are the following:

- The throughput of the communication channel and of the network devices is limited.
- The data delivery time from source to destination is non-zero.
- A communication channel is a medium with a set of physical parameters that can have influence on the signal propagation.
- The software and hardware architecture of the network devices impacts the way in which the data is being transmitted.

There are three main quality metrics:

- Losses.
- Delay.
- Jitter.

Let's look at each metric using an example: Node-2 transmits three data packets to Node-5; the data source and the recipient are connected to an intermediate Network device and the packets are part of the same service, i.e. their key service fields are the same.

Losses

During a data stream transmission, some packets may not be received, or may be received with errors. This process is called data loss and it is defined as the ratio between the number of received packets and the number of transmitted packets. In the example below (Figure 3), Node-2 transmits packets with the identifiers 1, 2 and 3, however, Node-5 receives only packets 1 and 3, i.e. the packet with the identifier 2 was lost. There are network mechanisms which allow the retransmission of the lost data. Examples of such mechanisms are the TCP and the ARQ protocols.

The causes of data loss can be divided into the following groups:

- **Losses in the medium:** losses related with the propagation of the signal in the physical environment. For example, the frame will be lost if the useful signal level is lower than the receiver sensitivity. Losses can also be caused by the physical damage of the interfaces connected to the media or by impulse pickups resulting from poor grounding.
- **Losses on the interface:** losses while processing a queue at the incoming or at the outgoing interface. Each interface has a memory buffer, which can be completely filled in case of intensive data stream transmissions. In this case, all the subsequent data entering the interface will be discarded, because it cannot be buffered.
- **Losses inside the device:** Data discarded by the network device according to the logic of the configuration. If the queues are full and the incoming data cannot be added to the processing queue, the network device will drop it. Also, these losses include the data packets rejected by access lists and by the firewall.

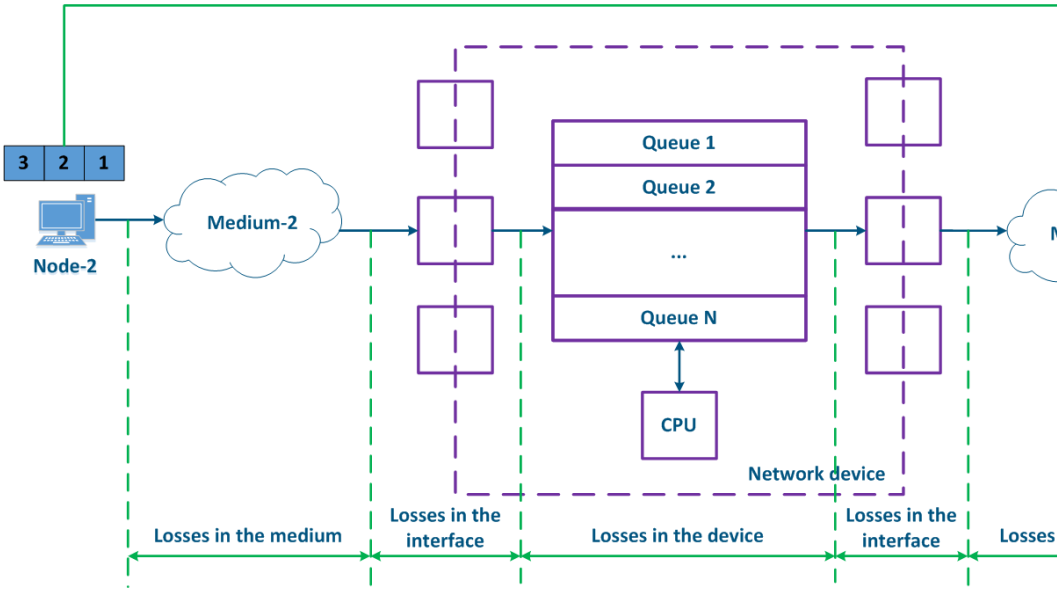


Figure 3 - Data packet loss example

The losses affect two indicators: throughput and packet performance.

Throughput

One of the main indicator that is practically used is the throughput, whose value depends on the losses. The throughput is defined by the capabilities of the physical channel and by the ability of the intermediate network devices to process the data stream. The link throughput is defined as the maximum amount of data that can be transmitted from the source to the receiver per unit of time.

Packet performance

The parameter that affects the throughput and the state of the queues is the packet performance of the device. Packet performance is the maximum number of data packets of a given length that a device is capable to process per unit of time.

The real throughput depends on both packet performance and on the interface's characteristics, therefore, at the network design stage, pay attention to the coherence of these parameters in order to avoid the situation when one of them becomes a bottleneck for a link or for a network segment.

The packet performance is defined by the hardware capabilities of the central processor and by the amount of internal memory. Network devices process multiple traffic streams with different L2 frame sizes, so the following Ethernet frame size values are used for a performance test:

- minimum size = 64 bytes;

- medium size = 512 bytes;
- maximum size = 1518 bytes.

Due to the limited amount of internal memory, better packet performance is achieved for the minimum frame size. Using minimum sized frames assumes a large amount of overhead since each data frame has a service header, whose size does not depend on the size of the frame itself.

For example, the service header length for 64 bytes long frames (Figure 4b) and 156 bytes frames(Figure 4c) will be the same, but the user data amount will be different. To transmit 138 bytes of user data, three 64 bytes frames or one 156 bytes frame will be required, so in the first case 192 bytes are sent, in the second only 156 bytes. For a link having a fixed throughput, large frames will increase the efficiency by rising the useful throughput of the system, but the latency will also increase. The performance of the Infinet devices in various conditions is shown in the "Performance of the Infinet Wireless devices" document.

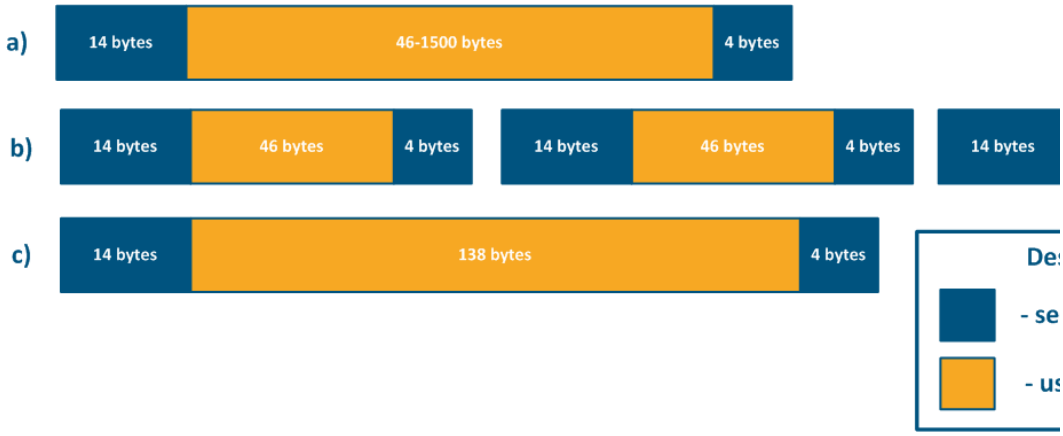


Figure 4 - Frame structure for various Ethernet frame lengths

Delay

Delay is defined as the time it takes for a packet to travel from the source to the destination. The value of the delay depends on the following aspects:

- **The signal's propagation duration in the medium:** depends on the physical characteristics of the medium and it is nonzero
- **Serialization time:** the conversion of a bitstream to a signal and backward by the incoming/outgoing interfaces is not instantaneous and makes use of the hardware resources of the network device.
- **Processing time:** the time spent by the data packet inside the network device. This time depends on the status of the packet queue, as a data packet will be processed only after processing the packets placed earlier in this queue.

The delay is often measured, as a round-trip time (RTT), i.e. the time it takes for the data packet to be transmitted from the source to the destination and backward. For example, this value can be seen in the ping command's results. The time it takes for the intermediate network devices to process the data packets forward and backward may differ, therefore, usually the round-trip time is not equal to the double of the one-way delay.

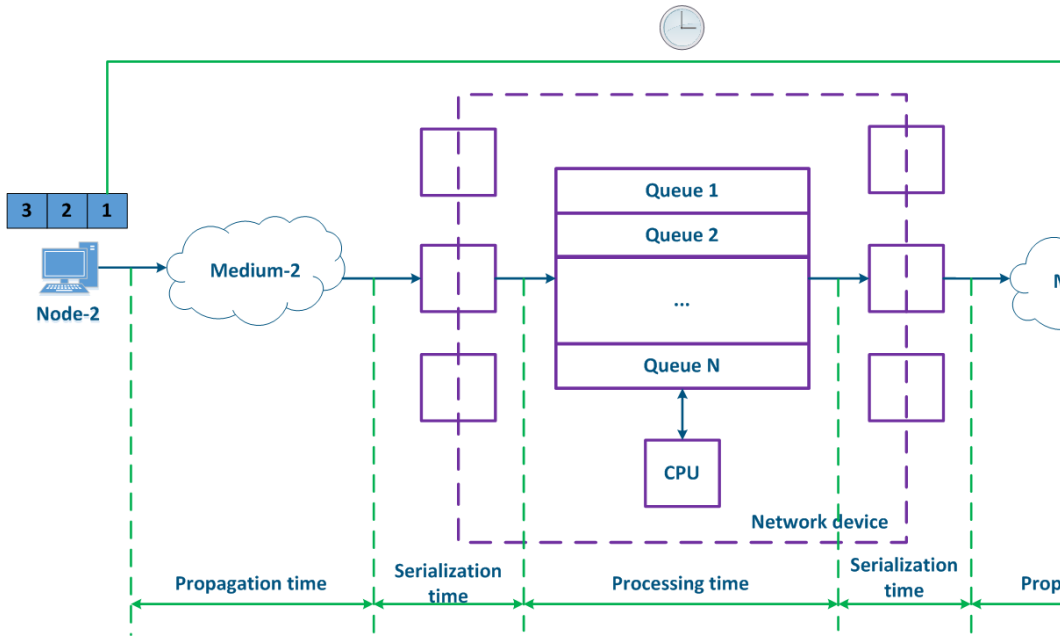


Figure 5 - Example of data transfer delay

Jitter

The CPU load and the status of the packet queues are frequently changing at the intermediate network devices, so the delay during the data packet transmission will vary. In the example below (Figure 6), the transmission time for the packets with the identifiers 1 and 2 is different. The difference between the maximum and the average delay values is called jitter.

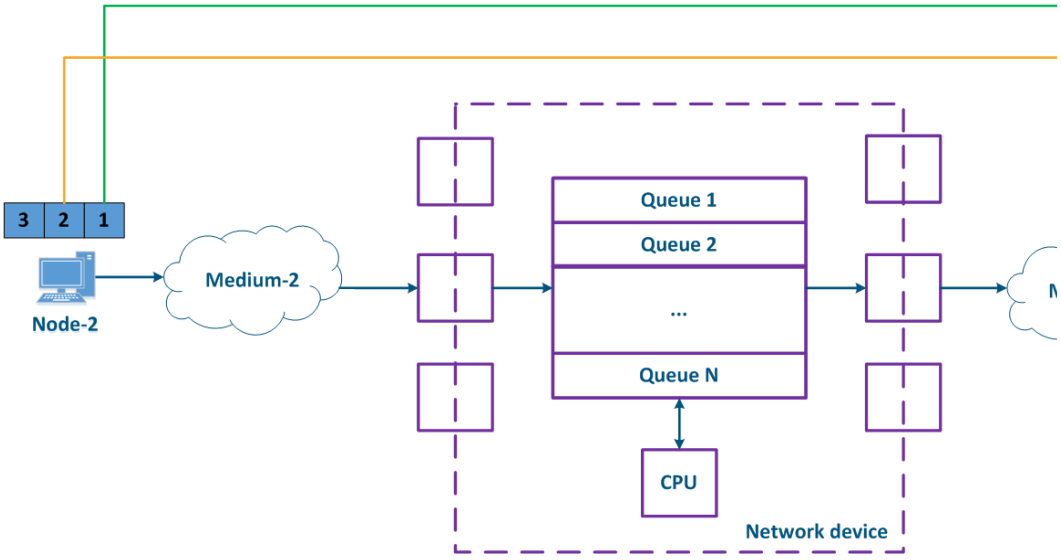


Figure 6 - Example of varying delay in data transfer

When using a redundant network infrastructure the data between the source and the receiver can be transmitted through different paths, so jitter will occur. Sometimes the difference between the delays on each path may become so large that the order of the transmitted data packets will change on the receiving side (Figure 7). In the example below, the packets were received in a different order.

The effect depends on the characteristics of the service and on the ability of the higher layer network protocols to restore the original sequence. Usually, if the traffic of different services is transmitted through different paths, then it should not affect the ordering of the received data.

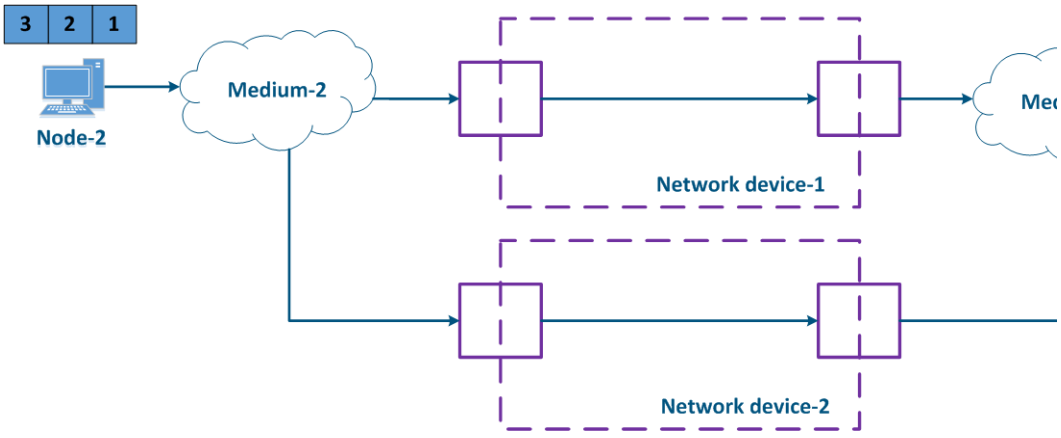


Figure 7 - Example of unordered data delivery

Service requirements with respect to the quality indicators

Each of the data transfer services has a set of requirements for the quality indicators. The RFC 4594 document includes the following service types:

Service	Indicator		
	Losses	Delay	Jitter
Network Control	low	low	low

Telephony	very low	very low	very low
Signaling	low	low	low
Multimedia Conferencing	medium	very low	low
Real-Time Interactive traffic	low	very low	low
Multimedia Streaming	medium	medium	low
Broadcast video	very low	medium	low
Low-Latency Data	low	medium	very low
Management	low	medium	medium
High-Throughput Data	low	high	high
Standard	undefined		
Low-Priority Data	high	high	high

Application Categories	Service Class	Signaled	Flow Behavior	G.1010 Rating
Application Control	Signaling	Not applicable	Inelastic	Responsive
Media-Oriented	Telephony	Yes	Inelastic	Interactive
	Real-Time Interactive	Yes	Inelastic	Interactive
	Multimedia Conferencing	Yes	Rate Adaptive	Interactive
	Broadcast Video	Yes	Inelastic	Responsive
	Multimedia Streaming	Yes	Elastic	Timely
Data	Low-Latency Data	No	Elastic	Responsive
	High-Throughput Data	No	Elastic	Timely
	Low-Priority Data	No	Elastic	Non-critical
Best Effort	Standard	Not Specified		Non-critical

QoS methods

The transmission of the various services is performed on a single network infrastructure, which has limited resources, therefore, mechanisms should be provided for distributing the resources between the services.

Let's look at the example below (Figure 8). Node-2 generates traffic serving different services with a total speed of 1 Gbit/s. Medium-2 allows to transfer this data stream to an intermediate network device, however, the maximum link throughput between the Network device and Node-5 is 500 Mbps. Obviously, the data stream cannot be processed completely and part of this stream must be dropped. The QoS task is to make these drops manageable in order to provide the required metric values for the end services. Of course, it is impossible to provide the required performance for all the services, as the throughput does not match, therefore, the QoS policy implementation involves that the traffic of the the critical services should be processed first.

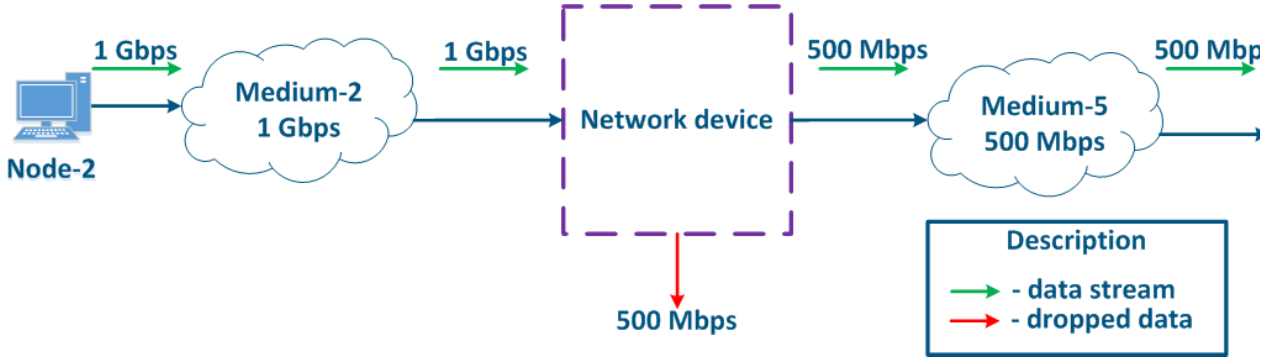


Figure 8 - Example of inconsistency between the incoming traffic amount and the link throughput

Two main methods used during the QoS policy implementation can be highlighted:

- **Prioritization:** the distribution of the data packets into queues and the extraction of the packets from the queues by their priority. In this case, the packets that are most sensitive to delay and jitter are processed first, then the traffic for which the delay value is not critical is processed.
- **Throughput limitation:** throughput limitation for the traffic flows. All the traffic that exceeds the set throughput threshold will be discarded.

Let's look at the example above, and add a second intermediate device to the data distribution scheme (Figure 9a). The packet distribution scheme follows the next steps:

- **Step 1:**
  - Node-1 and Node-2 generate packets for two services: telephony and mail. The telephony traffic is sensitive to delay and jitter unlike the mail service data (see Services requirements for quality indicators), therefore, it must be processed first by the intermediate devices.
  - Network device-1 receives the packets of Node-1 and of Node-2.
- **Step 2:**
  - Traffic prioritization is configured on Network device-1, thus the device classifies the incoming traffic and places the data packets in different queues. All the voice traffic will be put in queue 0, and the mail traffic will be put in queue 16. Thus, the priority of queue 0 is higher than the one of queue 16.
  - The packets leave the queues and proceed towards the outgoing interfaces in accordance with the queue priorities i.e. queue 0 will be emptied first, then queue 16 will be emptied.
- **Step 3:**
  - Network device-1 sends data to Medium-7, which is connected with Network device-2. The sequence of data packets corresponds to the quality metrics - the telephony data is transmitted first through the medium, and the mail service is sent next.
  - Node-3 is connected to Network device-2 and generates a mail service data stream.
- **Step 4:**
  - Network Device-2 has no prioritization settings, thus all the incoming traffic is put in queue 16. The data will leave the queues in the same order that it entered, i.e. the telephony and the mail services will be handled equally, despite the requirements of the quality indicators.
  - Network device-2 increases the delay for the telephony traffic transmission.
- **Step 5:**
  - The data is transmitted to the end nodes. The transmission time of the voice packets was also increased due to the additional processing of the mail service traffic of Node-3.

Each intermediate network device without traffic prioritization settings will increase the data transmission delay, so the value of the delay is unpredictable. Thus, having a large number of intermediate devices without QoS policies implemented, will make the real-time services's operation impossible because of the mismatch with the quality indicators, i.e. traffic prioritization must be performed along the entire network traffic transmission path (Figure 9b).

Keep in mind that implementing QoS policies is the only method to ensure the quality metrics. For an optimal effect, the QoS configuration should be synchronized with other settings. For example, using the TDMA technology instead of Polling on the InfiLINK 2x2 and InfiMAN 2x2 families of devices reduces jitter by stabilizing the value of the delay (see TDMA and Polling: Application features).

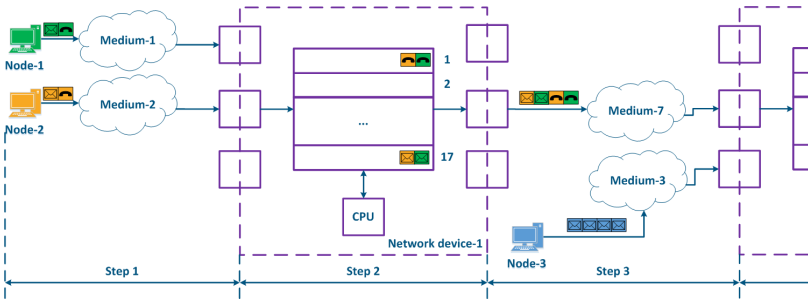


Figure 9a - Example of data distribution with partly imp

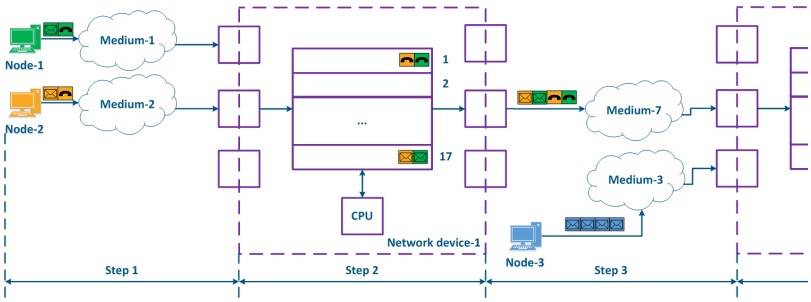


Figure 9b - Example of data distribution with implen

The traffic prioritization mechanism

From the management point of view, the transmission path through the network can be described in two ways (Figure 10a, b):

- **White-box:** all the network devices along the data propagation path are in the same responsibility zone. In this case, the QoS configuration on the devices can be synchronized, according to the requirements specified in the section above.
- **Black-box:** some network devices in the data propagation path are part of an external responsibility zone. The classification rules for incoming data and the algorithm for emptying the queues are configured individually on each device. The architecture of the packet queues's implementation depends on the manufacturer of the equipment, therefore there is no guarantee of a correct QoS configuration on the devices in the external responsibility zone, and as a result, there is no guarantee of the high-quality performance indicators.

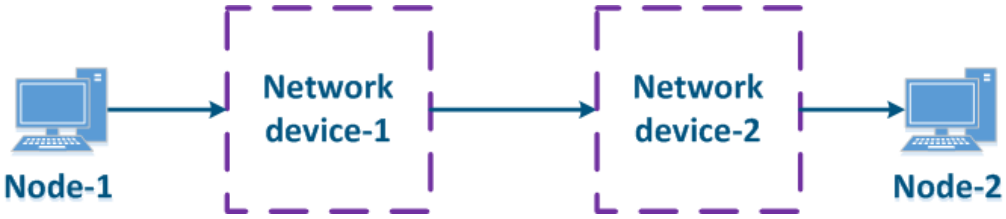


Figure 10a - White-box structure example

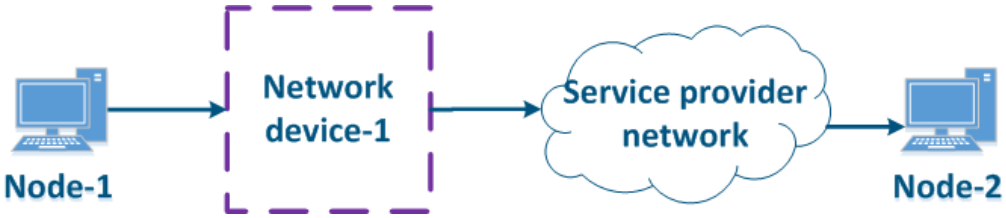


Figure 10b - Black-box structure example

To solve the described problem of the black-box network structure, the packet headers can be labeled: the priority required during packet processing is set in a header field and is kept over the whole path. In this case, all intermediate devices can put the incoming data in a queue according to the field values in which the priority is indicated. This requires the development of standard protocols and the implementation of these protocols by the equipment manufacturers.

Keep in mind that usually, the equipment located in an external responsibility zone does not support data prioritization in accordance with the priority values in the service headers. Traffic priority coordination should be performed at the border of the responsibility zones, at the administrative level, by implementing additional network configuration settings.

The processing priority of a packet can be set using the service fields of various network protocols. This article describes the use of the Ethernet and of the IPv4 protocol headers.

Ethernet (802.1p) frame prioritization

The Ethernet frame header includes the "User Priority" service field, which is used to prioritize the data frames. The field has a size of 3 bits, which allows to select 8 traffic classes: 0 - the lowest priority class, 7 - the highest priority class. Keep in mind that the "User Priority" field is present only in 802.1q frames, i.e. frames using VLAN tagging.

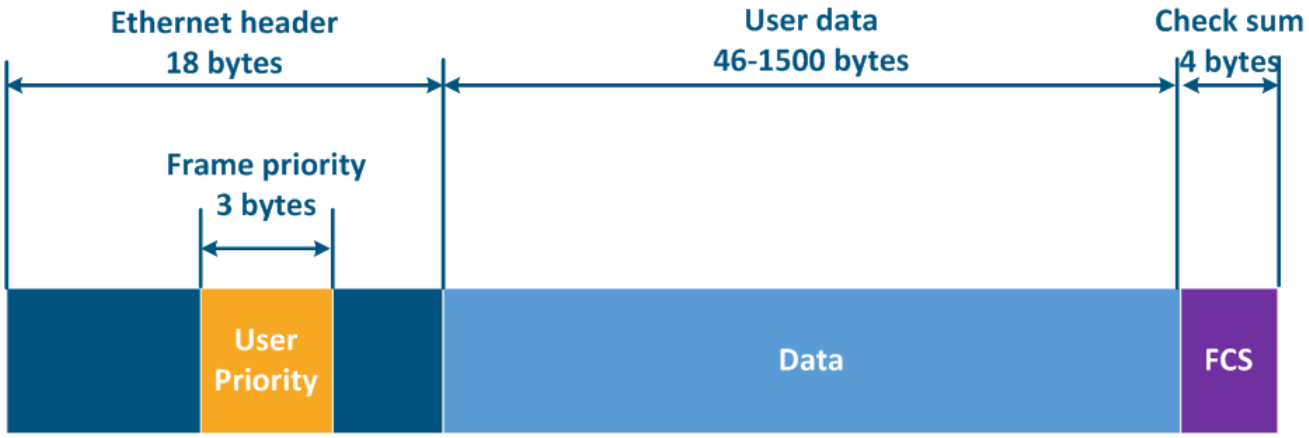


Figure 11 - Frame prioritization service field in the Ethernet header

IP packet prioritization

The IP protocol has three historical stages in the development of the service field responsible for packet prioritization:

1. When the protocol was first approved, there was an 8-bit ToS (Type of Service) field in the IP packet header (see RFC 791). ToS included the following fields (Figure 12a):
    a. Precedence: priority value (3 bits).
    b. Delay: delay minimization bit.
    c. Throughput: throughput minimization bit.
    d. Reliability: reliability maximization bit.
    e. 2 bits equal to 0.
2. In the second stage, 8 bits were still used for packet prioritization, however, ToS included the following fields (see RFC 1349):
    a. Delay.
    b. Throughput.
    c. Reliability.
    d. Cost: bit to minimize the cost metric (1 bit is used, whose value was previously zero).
3. Last, the IP header structure has been changed (see RFC 2474).The 8 bits previously used for prioritization were distributed in the following way (Figure 12b):
    a. DSCP (Differentiated Services Code Point): packet priority (6 bits).
    b. 2 bits are reserved.

Thus, ToS allows to distinguish 8 traffic classes: 0 - the lowest priority, 7 - the highest priority, and DSCP - 64 classes: 0 - the lowest priority, 63 - the highest priority.

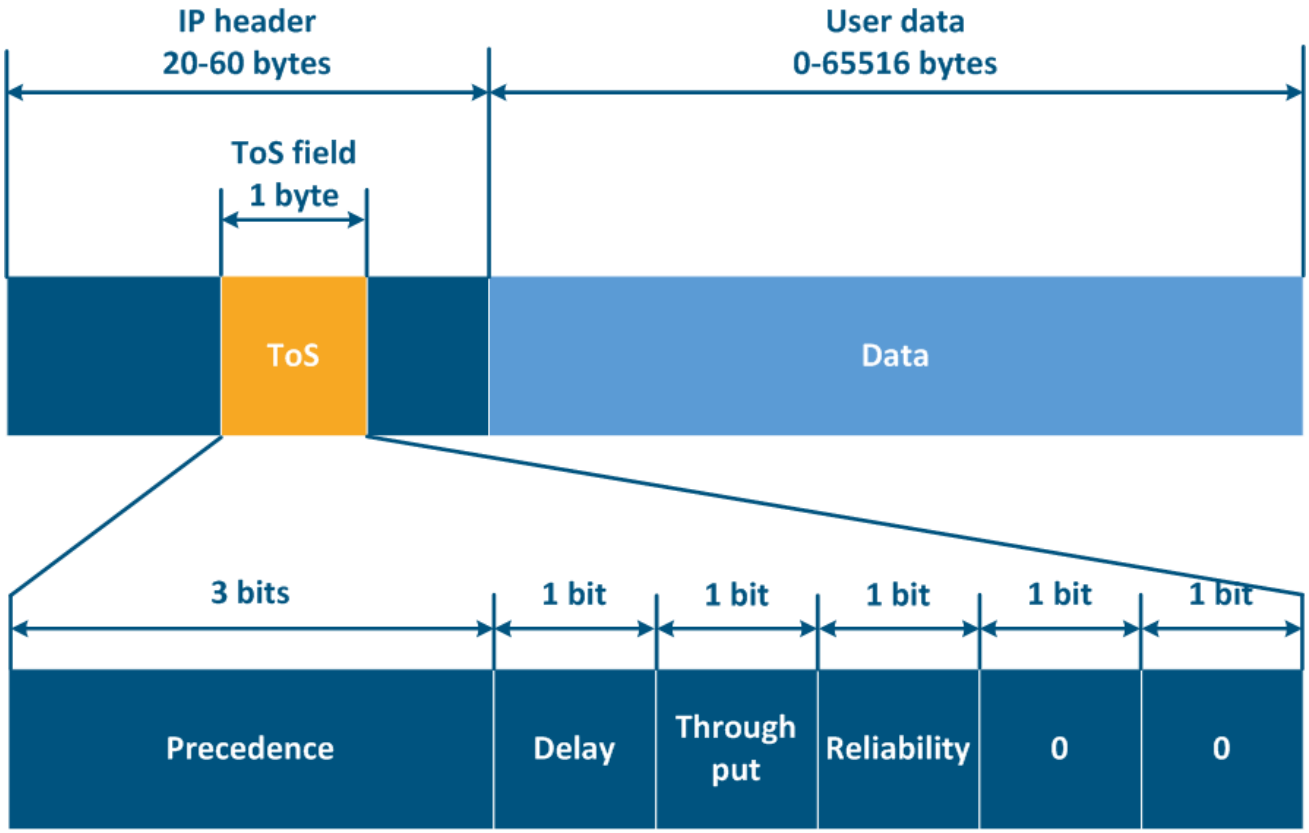


Figure 12a - ToS service field in the IP packet header

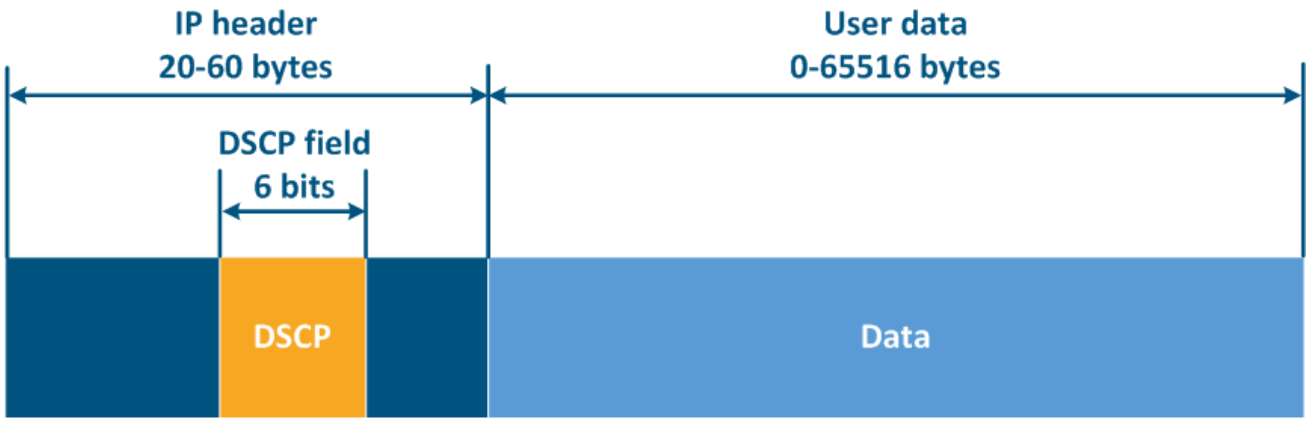


Figure 12b - DSCP service field in the IP packet header

Priority configuration

Many end nodes in the network do not support the handling of the service headers: can not set or remove the priority, so this functionality should be implemented on the corresponding intermediate network devices.

Let's look at the example of a data transmission from Node-1 to Node-2 through a DS-domain and through a third-party telecom operator's network (Figures 13a-c). The DS domain includes three devices, two of them are located at the borderline and one is an intermediate device. Let's look at the steps taken for data processing in a network using an Ethernet frame transmission (the basic principles discussed in the example below are applicable for an IP packet or other protocol that supports data prioritization):

- **Step 1:** Node-1 generates an Ethernet frame for Node-2. There is no field present for frame priority tagging in the header (Figure 13a).

- **Step 2:** Border Network Device-1 changes the Ethernet header, setting the priority to 1. Border devices should have rules configured in order to filter the traffic of Node-1 from the general stream and to assign a priority for it. In networks with a large traffic flow number, the list of rules on border devices can be volumetric. Border network device-1 processes the frame according to the set priority, placing it in the corresponding queue. The frame is transmitted towards the outgoing interface and sent to Intermediate network device-2 (Figure 13a).
- **Step 3:** Intermediate network device-2 receives the Ethernet frame having priority 1, and places it in the corresponding priority queue. The device does not handle the priority in terms of changing or removing it inside the frame header. The frame is next transmitted towards the Border network device-3 (Figure 13a).
- **Step 4:** Border network device-3 processes the incoming frame similarly to the Intermediate device-2 (see Step 3) and forwards it towards the service network provider(Figure 13a).
  - **Step 4a:** in case of agreeing that the traffic will be transmitted through the provider's network with a priority other than 1, then Border Device-3 must change the priority. In this example, the device changes the priority value from 1 to 6 (Figure 13b).
- **Step 5:** during the transmission of the frame through the provider's network, the devices will take into account the priority value in the Ethernet header (Figure 13a).
  - **Step 5a:** similarly to Step 4a (Figure 13b).
  - **Step 5b:** if there is no agreement about the frame prioritization according to the priority value specified in the Ethernet header, a third-party service provider can apply its own QoS policy and set a priority that may not satisfy the QoS policy of the DS domain (Figure 13c).
- **Step 6:** the border device in the provider's network removes the priority field from the Ethernet header and forwards it to Node-2 (Figure 13a-c).

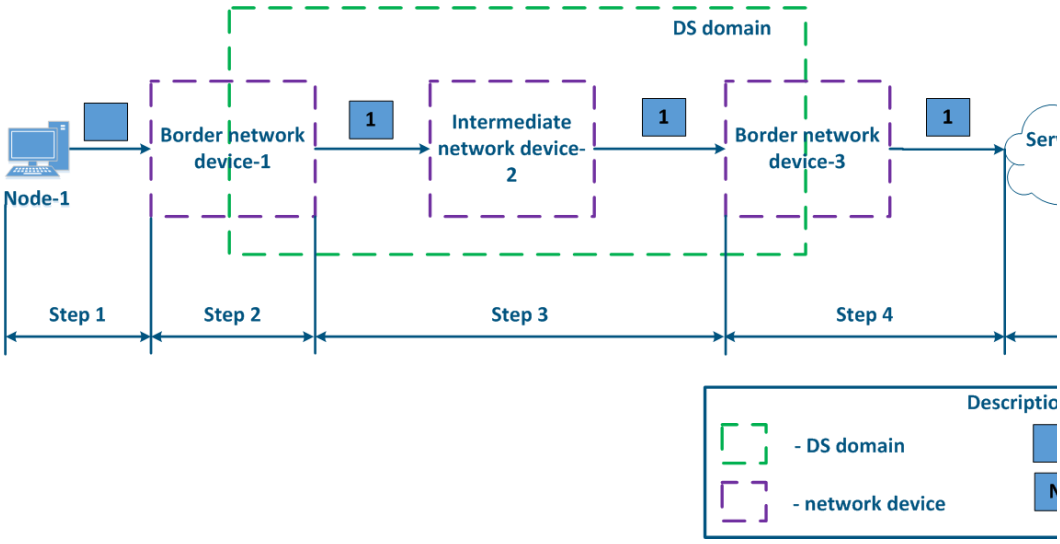


Figure 13a - Example of Ethernet frame priority changing during the transmission through two network segments (the priority setting is coordinate

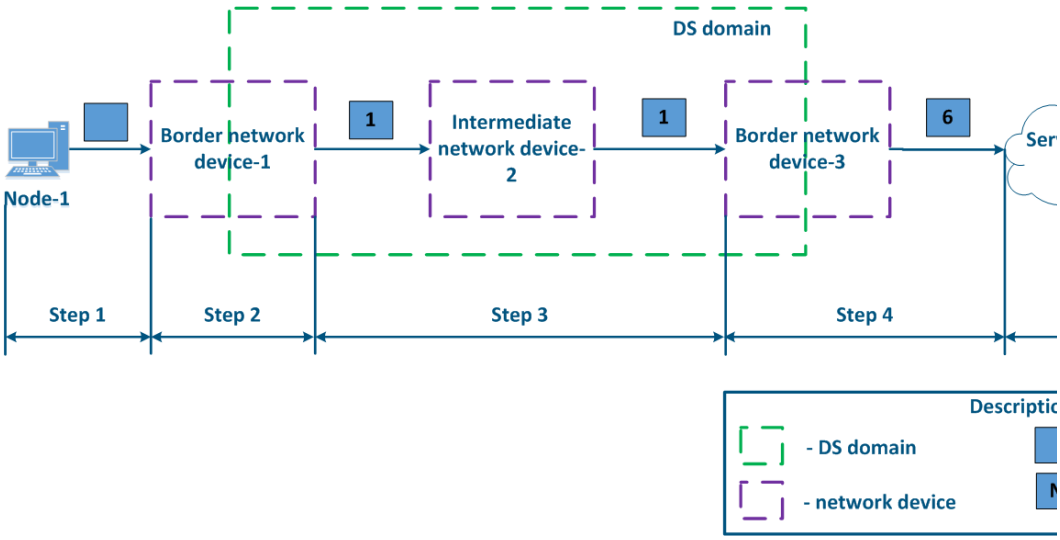


Figure 13b - Example of Ethernet frame priority changing during the transmission through two network segments (the priority setting is coo

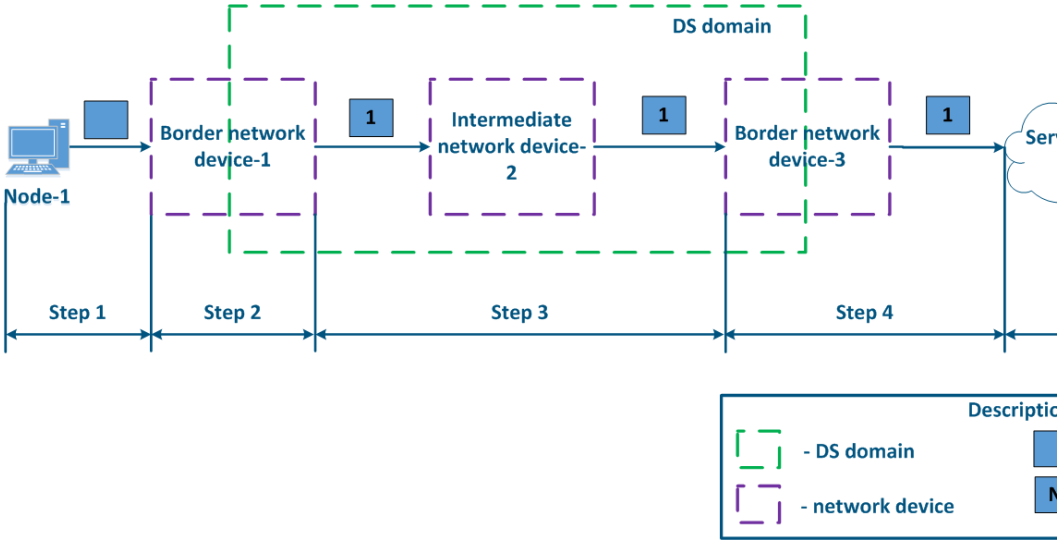


Figure 13c - Example of Ethernet frame priority changing during the transmission through two network segments (the priority setting

Queues implementation in Infinet devices

For a device, the process of analyzing the priority in the service headers and the data processing according to these priorities is not a simple task due to the following reasons:

- The devices automatically recognize priorities according to different protocols. For example, the InfiLINK XG family of devices supports 802.1p prioritization, but does not recognize DSCP priority values.
- The devices at the borderline of the DS domain allow to use a different set of criteria to classify the traffic. For example, the InfiMAN 2x2 devices allow to set priorities by selecting all the TCP traffic directed to port 23, while the Quanta 5 family devices does not support this type of prioritization.
- The number of the queues implemented inside the devices differs and depends on the manufacturer. A correspondence table is used to set a relation between the priority in the service header and the device's internal queue.

The tables below show the data types for the queues of the internal architecture, the priority handling possibilities and the relation between the standardized priorities and the internal priorities used by the device.

Please note the architectural queuing feature of the Infinet devices: all queues share a single memory buffer. In case that all the traffic falls into a single queue, the size of the queue will be equal to the size of the buffer, but if there will be several queues in use, the size of the memory buffer will be evenly divided between them.

Internal packet queuing

Parameter	Description	InfiLINK 2x2 / InfiMAN 2x2	InfiLINK Evolution / InfiMAN Evolution	InfiLINK XG / InfiLINK XG 1000	Quanta 5 / Quanta 6 / Quanta 70
Marking criteria	A criteria that can be used to classify the incoming traffic.	PCAP expressions support (PCAP expressions allow flexible filtering based on any service header field, see the PCAP filters article)	PCAP expressions support (PCAP expressions allow flexible filtering based on any service header field, see the PCAP filters article)	vlan-id	vlan-id
Auto recognition	Protocols for which the family of devices automatically recognizes the priority set in the header and puts the data in the appropriate queue.	RTP 802.1p IPIP/GRE tunnels MPLS DSCP ToS ICMP	RTP 802.1p IPIP/GRE tunnels MPLS DSCP ToS ICMP	802.1p	802.1p

		TCP Ack PPPoE	TCP Ack PPPoE		
Number of queues	The number of data queues used by the device.	17	17	4	8
Queue management	Supported mechanisms for emptying the packets from the queues.	Weighted		Strict Weighted	
QoS configuration via Web	Documentation about the traffic prioritization configuration using the Web interface.	QoS options Traffic Shapping	QoS options Traffic Shaping	Configuring QoS Switch Configuring per-VLAN	Switch settings
QoS configuration via CLI	Documentation about the traffic prioritization configuration using the command line interface.	qm command	qm command	Commands for switch configuration	-

Correspondence between the priorities of the standard protocols and the internal priorities used by the InfiLINK 2x2, InfiMAN 2x2, InfiLINK Evolution and InfiMAN Evolution families of devices

Traffic class (in accordance with MINT)	InfiLINK 2x2, InfiMAN 2x2, InfiLINK Evolution and InfiMAN Evolution	802.1p	ToS (Precedence)	DSCP
Background	16	01		
Regular best effort	15	00	00	0
Business 6	14		01	8, 10
Business 5	13			12, 14
Business 4	12		02	16, 18
Business 3	11			20, 22
Business 2	10		03	24, 26
Business 1	9	02		28, 30
QoS 4	8		04	32
QoS 3	7			34
QoS 2	6			36
QoS 1	5	03		38
Video 2	4	04	05	40, 42
Video 1	3			44, 46
Voice	2	05	06	48, 50
Control	1	06		52, 54
NetCrit	0	07	07	56, 58, 60, 62

Correspondence table between the priorities of the standard protocols and the internal priorities used by the InfiLINK XG, InfiLINK XG 1000, Quanta 5, Quanta 6 and Quanta 70 families of devices

Traffic class (in accordance with 802.1p)	802.1p	InfiLINK XG, InfiLINK XG 1000	Quanta 5, Quanta 6, Quanta 70
Background (lowest priority)	00	1	0
Best Effort	01		1
Excellent Effort	02	2	2
Critical Applications	03		3
Video	04	3	4
Voice	05		5

Internetwork Control	06	4	6
Network Control (higher priority)	07		7

Queue management

Prioritization assumes the use of several packet queues, whose content must be transmitted to the outgoing interfaces through a common bus. Infinet devices support two mechanisms for packet transmission from the queues to the bus: strict and weighted scheduling.

Strict scheduling

The strict prioritization mechanism assumes a sequential emptying of the queues according to the priority values. Packets with priority 2 will only be sent after all the packets with priority 1 have been transferred to the bus (Figure 14). After the packets with priorities 1 and 2 are sent, the device will start sending packets with priority 3.

The disadvantage of this mechanism is that resources will not be allocated to low-priority traffic if there are packets in the higher priority queues, leading to the complete inaccessibility of some network services.

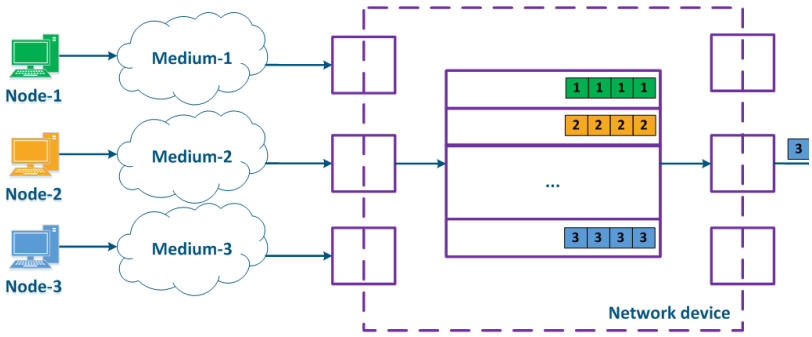


Figure 14 - Strict scheduling

Weighted scheduling

The weighted scheduling doesn't have the disadvantages of the strict scheduling. Weighted scheduling assumes the allocation of the resources for all the queues according to the weighting factors that correspond to the priority values. If there are three queues (Figure 15), weighted factors can be distributed in the following way:

- packet queue 1: weight = 3;
- packet queue 2: weight = 2;
- packet queue 3: weight = 1.

When using the weighted scheduling, each queue will receive resources, i.e. there will be no such situation with complete inaccessibility of a network service.

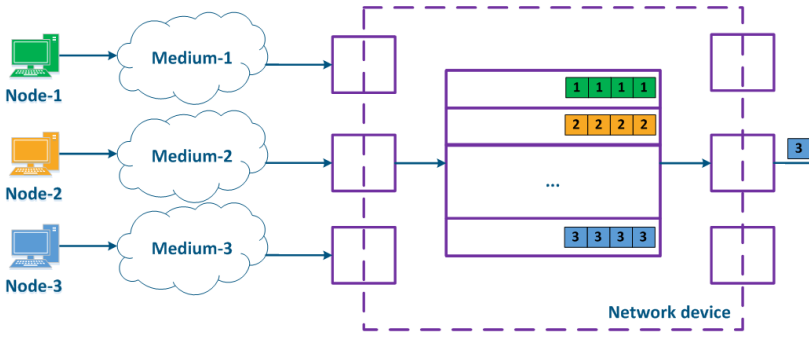


Figure 15 - Weighted scheduling

Traffic prioritization recommendations

Universal recommendations for configuring traffic prioritization mechanisms:

- Pay special attention when developing the QoS policies. The policy should take into account the traffic of all the services used in the network and it should provide strict compliance between the service and the traffic class.

- The QoS policy should take into account the technical capabilities of the devices for recognizing and handling the service field values, which indicate the data priority.
- The rules for traffic flow classification must be configured on the border devices of the DS domain.
- The intermediate devices of the DS domain should automatically recognize the traffic priorities.

Throughput limitation mechanism

The distribution of the network resources between the traffic flows can be performed not only by prioritization, but also using the throughput limitation mechanism. In this case, the bitrate of the stream cannot exceed the threshold level set by the network administrator.

The speed limitation principle in Infinet devices

The throughput limitation principle is to constantly measure the throughput of the data stream and to apply restrictions if the this value exceeds the set threshold (Figure 16a,b). The throughput limitation in Infinet devices is performed according to the Token Bucket algorithm, where all data packets above the throughput threshold are discarded. As a result, there will appear losses, as described above.

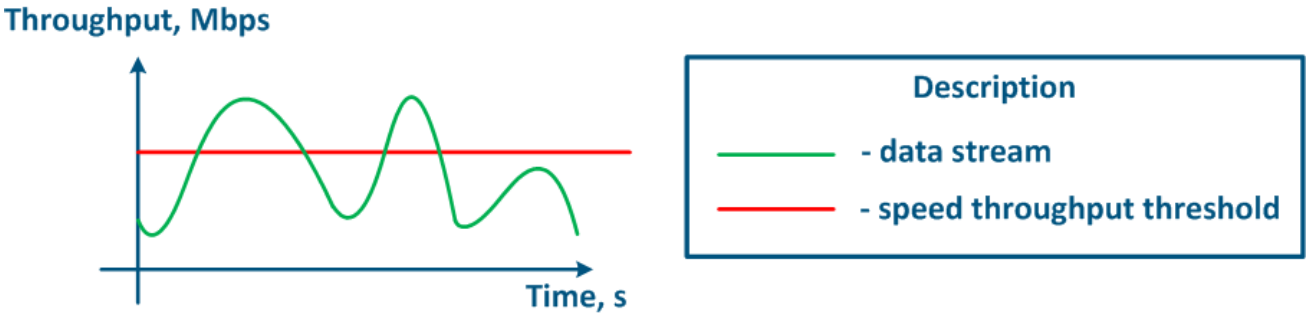


Figure 16a - Unlimited data flow rate

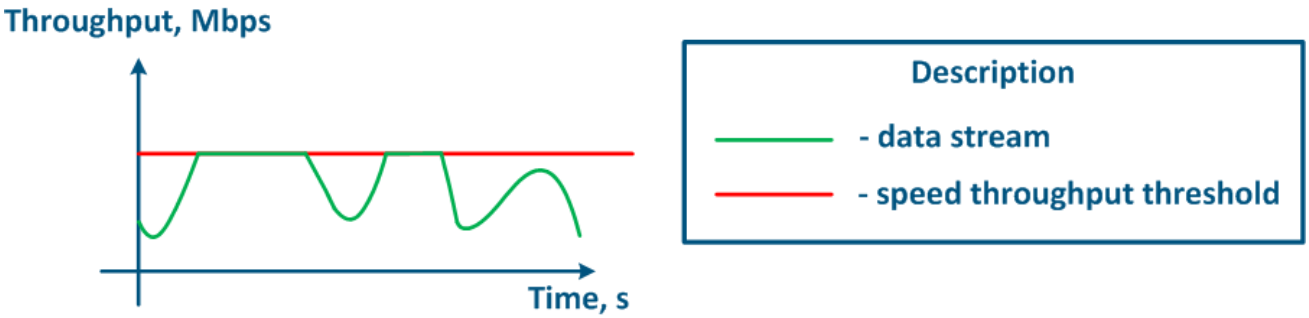


Figure 16b - Limited data flow rate

Token Bucket Algorithm

For each speed limit rule there is a logical buffer associated, in order to serve the allowed amount of transmitted data. Usually, the buffer size is larger than the size of the limitation. Each unit of time is allocated a data size equal to the threshold for the bitrate limit.

In the example below (video 1), the speed limit is 3 data units and the buffer size is 12 data units. The buffer is constantly filled according to the threshold, however, it cannot be filled over its own size.

Your browser does not support the HTML5 video element

Video 1 - Resource allocation into a speed limit buffer

The data received by the device at the inbound interface will be processed only if the buffer has resources for their processing (video 2). Thus, the passing data occupies the buffer's resources. If the buffer's resources are fully occupied at the time of a new data frame arrival, the frame will be discarded.

Your browser does not support the HTML5 video element

Video 2 - Dedicated resources usage for data processing

Keep in mind that the resource allocation and the data processing are performed simultaneously inside the buffer (video 3).

The rate of the data flows in packet networks is inconsistent, proving the efficiency of the Token Bucket algorithm. The time intervals in which data is not transmitted allows to accumulate resources in the buffer, and then process the amount of data that exceeds the threshold. A wide band will be allocated to pulse data streams, such as web traffic, in order to ensure a quick loading of the web pages and to increase the comfort level of the end user.

Besides the described advantage of the Token Bucket algorithm, the average throughput will match with the set threshold, as in a long period of time, the amount of available resources will be determined not by the size of the buffer, but by the intensity of its filling, which is equal to the throughput threshold.

Your browser does not support the HTML5 video element

Video 3 - Data processing at the speed limit buffer

The Token Bucket algorithm can be applied to separate traffic flows. In this case, a speed limit buffer will be allocated for each flow (video 4).

In this example, two speed limit rules are implemented: for the traffic of vlan 161 - 3 data units per unit of time, for the traffic of vlan 162 - 2 data units. The buffer size for each traffic flow contains 4 time intervals, i.e. 12 data units for vlan's 161 traffic and 8 data units for vlan's 162 traffic. In total, 5 data units are allocated to the buffers in each time interval, then the allocated resources are distributed between the buffers. Since the size of the buffers is limited, the resources that exceed their size cannot be used.

Your browser does not support the HTML5 video element

Video 4 - Resource allocation for two speed limit buffers

Each buffer's resources can only be used for the traffic of the corresponding service (video 5). Thus, to handle vlan's 161 traffic, only the resources of the buffer for vlan's 161 traffic are used. Similarly, the other buffer's resources are used for vlan's 162 traffic.

Your browser does not support the HTML5 video element

Video 5 - Usage of the dedicated resources for data processing

There are ways to combine the resource buffers. For example, on the Infinet devices, the allocated resource buffers can be combined using classes (see below). If one resource buffer is filled with resources (video 6), its further incoming resources can be provided to another buffer.

In the example below, the buffer for vlan 162 is full of resources, allowing to fill in the vlan's 161 buffer with 5 data units of resources, instead of 3 (its own 3 data units plus the 2 data units of the other buffer). In this case, the vlan's 161 service throughput will increase. But when vlan's 162 traffic resource buffer will have free space, the resource allocation will return to the normal mode: for vlan's 161 buffer - 3 data units, for vlan's 162 buffer - 2 data units.

Your browser does not support the HTML5 video element

Video 6 - Redistribution of the allocated resources between various speed limit buffers

Types of speed limits in Infinet devices

The throughput limitation principle described above is implemented in the Infinet devices in two ways:

- Traffic shaping at the physical interface: limitations will be applied to the whole data flow passing through the physical interface. This method is easy to configure - specify the interface and the threshold value - but it does not allow to apply limitations to a specific network service traffic.
- Traffic flow shaping: limitations are applied to the logical data flows. The logical data stream is separated from the main traffic by a specified criteria. It allows to apply throughput limitations per network services, which are separated by the values of the service header fields. For example, the traffic tagged with vlan 42 can be separated to a logical channel and limited in throughput without influencing the other traffic flows.

The Infinet devices allow to configure hierarchical throughput allocation structures. Two object types are used to perform this: a logical channel and a class, which are connected by a child-parent relationship. The class has a throughput value assigned, which is distributed between the child logical channels, and the channel has a guaranteed and a maximum throughput value - CIR and MIR.

Let's look at the example of transmitting the traffic of two services associated with vlan id's 161 and 162, between Master and Slave (Figure 17a). The total traffic of the services should not exceed 9 Mbps.

The Master's device configuration can be performed in the following way (Figure 17b):

- Class 16 has been configured with a 9 Mbps throughput.
- Class 16 is the parent of the channels 161 and 162, i.e. the total traffic at these logical channels is limited to 9 Mbps.
- The traffic with vlan ID 16 is associated with the logical channel 161; the traffic of vlan 162 is associated with the logical channel 162.
- The CIR value for channel 161 is 4 Mbps and for channel 162 it is 5 Mbps. If both services will actively exchange data, the threshold values for their traffic will be equal to the CIR of each channel.
- The MIR value for channel 161 is 9 Mbps and for channel 162 it is 7 Mbps. If there is no traffic in logical channel 162, then the threshold value for channel 161 will be equal to the MIR, i.e. 9 Mbps. In the other case, when there is no traffic in the logical channel 161, the threshold value for channel 162 will be equal to 7 Mbps.

Title

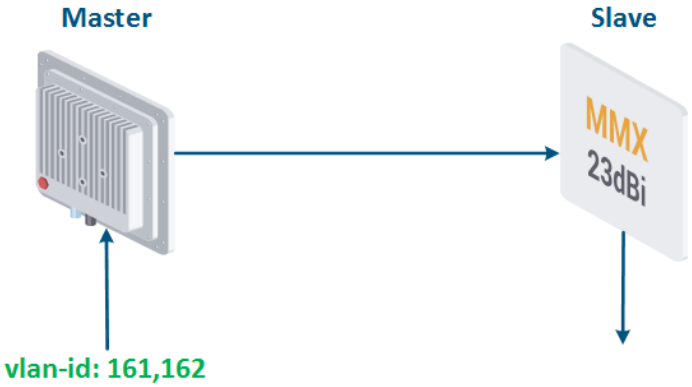
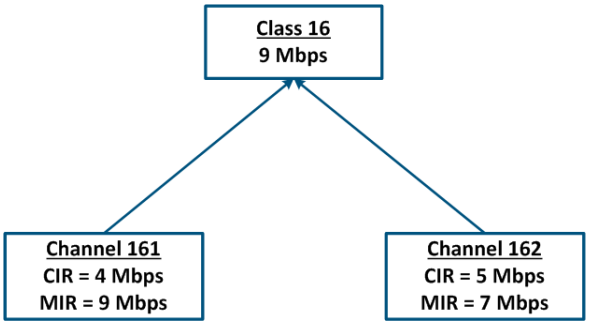


Figure 17a - Throughput limitation for 2 traffic flows tagged with vlan-ids 161 and 162



Scenario	vl
• vlan 161 traffic: present • vlan 162 traffic: present	4
• vlan 161 traffic: present • vlan 162 traffic: absent	9
• vlan 161 traffic: absent • vlan 162 traffic: present	

Figure 17b - Hierarchical channel structure of the throughput limits for the traffic of vlans 161 and

The throughput limitation capabilities of all Infinet families of devices are shown in the table below:

Throughput limitation capabilities in Infinet devices

Parameter	Description	InfiLINK 2x2 / InfiMAN 2x2	InfiLINK Evolution / InfiMAN Evolution	InfiLINK XG / InfiLINK XG 1000
Interface shaping	The throughput limitation capabilities of the device's physical interface.	-	-	• GE0 • GE1 • SFP • mgmt
Logical stream shaping	The throughput limitation capability for a traffic stream, filtered according to one or more criteria.	up to 200 logical channels	up to 200 logical channels	-
Traffic directions	Ability to apply limitations to the incoming/outgoing traffic flows.	incoming and outgoing	incoming and outgoing	outgoing
Limitations hierarchy	The ability to create a system of hierarchical limitations.	up to 200 logical channels, which are the children of the logical classes	up to 200 logical channels, which are the children of the logical classes	-
Logical stream filtering	Criteria used to filter the data streams.	PCAP expressions support (PCAP expressions allow to perform a flexible limitation based on any service header field, see the PCAP filters article)	PCAP expressions support (PCAP expressions allow to perform a flexible limitation based on any service header field, see the PCAP filters article)	-
Traffic shaping in Web	Documentation about throughput limitation settings in the Web interface.	Traffic shaping	Traffic Shaping	Switch
Traffic shaping in CLI	Documentation about throughput limitation settings via CLI.	qm command	qm command	Commands for switch

				configuration

Recommendations for the throughput limitation configuration

Use the following recommendations during the data throughput limitation configuration:

- The traffic of all network services should be limited. It allows to take control over all traffic flows and separately allocate resources for these flows.
- The throughput limitation should be performed on the devices closest to the data source. There is no need to duplicate throughput limiting rules for the data flows throughout the chain of intermediate devices.
- Many network services are bidirectional, so they require restrictions on devices for both the incoming and the outgoing traffic.
- To set the correct throughput threshold values, evaluate first the average and the maximum values of the service traffic. Pay special attention to the busy hours. Collecting data for analysis is possible via the InfiMONITOR monitoring system.
- The sum of the CIR values of the logical channels associated with one class should not exceed the maximum class throughput.

Additional materials

White papers

1. TDMA and Polling: Application features.
2. Performance of the Infinet Wireless devices.

Webinars

1. QoS policies configuration in Infinet Wireless devices.

Videos

1. Quality of Service With Infinet Wireless Units.

Others

1. RFC 4594.
2. RFC 791.
3. RFC 1349.
4. RFC 2474.
5. InfiMONITOR monitoring system.
6. InfiLINK 2x2, InfiMAN 2x2 family devices web interface. QoS options.
7. InfiLINK 2x2, InfiMAN 2x2 family devices web interface. Traffic shaping.
8. InfiLINK Evolution, InfiMAN Evolution family devices web interface. QoS options.
9. InfiLINK Evolution, InfiMAN Evolution family devices web interface. Traffic shaping.
10. InfiLINK XG, InfiLINK XG 1000 family devices web interface. Configuring QoS.
11. Quanta 5, Quanta 6 family devices web interface. Switch settings.
12. Quanta 70 family devices web interface. Switch settings.
13. QoS configuration in OS WANFleX.