QoS in ATM Networks

Layered model

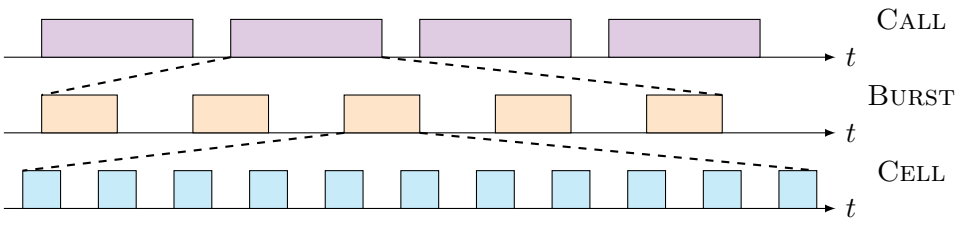
In order to define QoS parameters and traffic characterization a layered model is defined; the following classes are introduced:

. call level;

. burst level;

. cell level.

The picture reported below show this division; notice that time scales are very different.



Call level

The call level is the highest level and the traffic, during a call, occupies network resources for the entire duration. Since this approach is end to end, QoS is provided at this level by control plane which is the plane who runs signalling; moreover, one of QoS parameters is the blocking probability, implemented point to point. Traffic is characterized by:

. call attributes;

. call model.

Call attributes are parameters needed to be specified during signalling; they are many, so the signalling step is huge for time and resources points of view. Here is reported the list of them:

. type of request (on demand, permanent, semi-permanent);

. configuration (point to point, multipoint, broadcast, *Lan emulation*);

. number of connections open in two directions;

. VPC/VCC;

. traffic contract element for each connection;

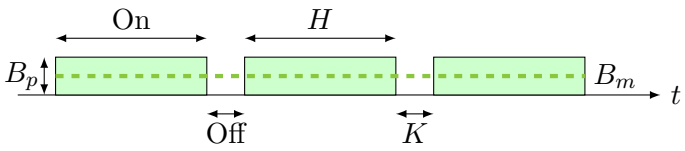
. signalling protocol used at network ingress;

. supplementary sercices.

The traffic characterization (call arrival process and call duration) is realized with a stochastic approach.

Burst level

Bursts are created by the packet fragmentation process: it is not much used and, for this level, no QoS parameters are defined. Traffic generation is done at the peak rate and it is *on/off*: *on* when packet is sent and *off* for silence.



Off periods and burst lenght are stochastic parameters; definition of other parameters can be ascribed of these two:

. burstiness:

$$\beta = \frac{H+K}{H} \frac{\text{total period}}{\text{active period}}$$

. activity coefficient:

$$\alpha = \frac{1}{\beta}$$

. average bit rate:

$$B_m = \alpha \cdot B_p$$

. bit rate variance:

$$\sigma_B^2 = B_m \cdot (B_p - B_m)$$

Cell level

The cell level is the lowest level; the traffic is characterized by:

. inter-arrival time distribution: it is complex but very complete;

. distribution of the number of cells generated in a measurement period T.

Often less information is required to reduce complexity:

. inter-arrival expected value and variance;

. the average bit-rate computed starting from the average inter-arrival time.

QoS parameters are provided for reliability and cell delay; in terms of reliability:

. cell loss probability;

. cell error probability (Hec: header error control);

. cell mis-insertion probability (errors on lables which implie that a cell belongs to the wrong VC);

while in terms of cell delay:

. expected value;

. variance;

. maximum.

Standard

The traffic contract is simple and is composed by:

. traffic characterization;

. QoS guarantee.

Traffic characterization

Main issues that characterized traffic are:

. identification of a cell flow within a connection in order to check if it is conformant;

. definition of traffic: nominal characteristics (traffic generate by users) and interfering traffic (management traffic);

. tolerance: due to multiplexing nominal characteristics can change and this parameter takes care of this issue with CDVT (Cell Delay Variation Tolerance);

. definition of conformant traffic with GCRA (Generic Cell Rate Algorithm), which is a chain of token bucket algorithms; non conformant traffic should become low priority traffic.

Users can generate traffic on a cell base, which is the lowest level in the layered model; cell over which confermance algorithms are run, can be divided into:

. aggregated flow (all kind of cells);

. data cells (no cells generate by switches: OAM, RM);

. high priority data cells;

. OAM cells;

. RM cells;

. data and OAM cells;

. high priority data and OAM cells.

According to traffic intrisic parameters it is possible distinguish:

. PCR (Peak Cell Rate), a worst case definition;

. SCR (Sustainible Cell Rate), an average definition: number of cells divided by the connection duration;

. IBT (Intrinsic Burst Tolerance), the maximum admissible advance time tolerate in the real process;

. MBS (Maximum Burst Size), the maximum burst which is transmit at peak.

They are negotiable parameters and, they are the minimum number of parameters declared by users. The last two parameters are defined as:

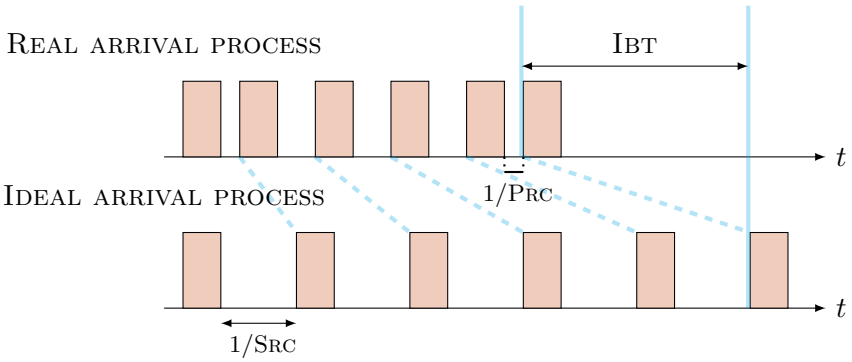
. IBT:

$$\text{IBT} = (\text{MBS} - 1) \cdot \left(\frac{1}{\text{SCR}} - \frac{1}{\text{PCR}} \right)$$

. MSB:

$$\text{MSB} = \frac{(1 + \text{IBT})}{\left(\frac{1}{\text{SCR}} - \frac{1}{\text{PCR}} \right)}$$

That formulas tells that, once one of the two is defined, the other comes for free.



Ibt, in the picture, gives flexibility at real traffic generation, but impose a constraint: a cell arrived earlier than it should be is not conformant.

GCRA

This algorithm is used to verify confermance with respect to a parameter: the time; it can run as a policer or as a shaper (adapting traffic: delay packets); as parameters, it receives:

. T: nominal cell inter-arrival time; it is defined as

$$\mathrm{T} = \frac{1}{\mathrm{Src}}$$

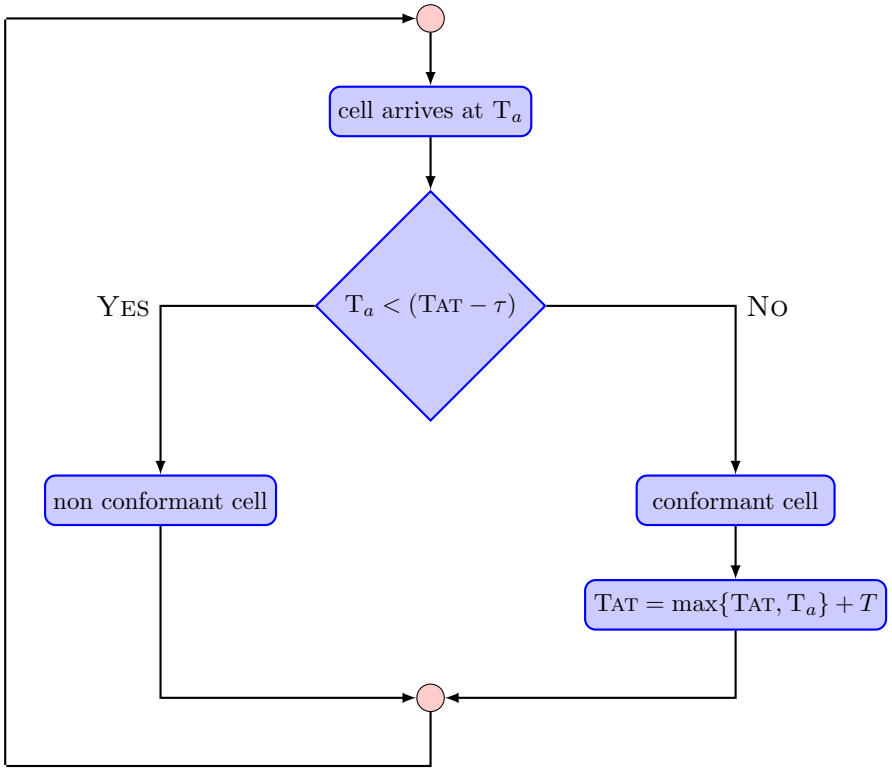
. τ: tolerance or maximum accepted variation vieth respect to the nominal space; it is used for non conformant cells, so cells who arrive earlier than they should be (cells who arrive later are always conformant).

As variables GCRA uses:

. T_a: real cell arrival time;

. Tat: theoretical arrival time.

Flow diagram In the following page is show the flow diagram representing passages exploit by GCRA.

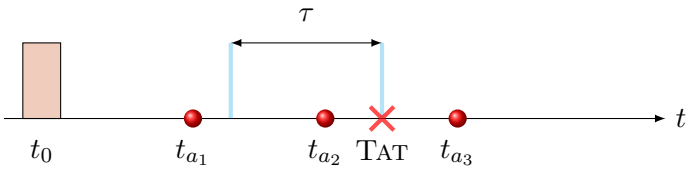


Example The first cell arrives at time 0: all possible arrivals of the second cell are:

. before than $TAT - \tau$;

. before than TAT;

. after TAT.



The only packet which is not conformat is t_{a_1}: it can be dropped or marked as a low priority packet. Now, consider TAT; how is it updated? There are, as before, three cases:

. if the cell is arrived in t_{a_1} there are no problems: no update is required since that cell disappear;

. if the cell is arrived in t_{a_2} the update can be done or starting from t_{a_1} or starting from TAT: the second case is the one implemented (the worst case) because the tolerance can be gained only once; if the update starts from t_{a_2}, each time, the tolerance (TAT $- t_{a_2}$) is preserved and the increase, instead being T, will be $(T - \tau)$;

. if the cell is arrived in t_{a_3} the issue is similar to the previous case: the update can be done starting from t_{a_3} or TAT; in this case, is again considered the worst case, which is TAT, otherwise the user gains a lot of credits if the has been silent for a long period: time lost can not be recoreverd.

Since multiplexing stages modify the original shape of the traffic (it implies that the delay will be unpredictable) CDVT (Cell Delay Variation Tolerance) try to compensate this traffic fluctuation. It is similar to IBT, but it does not allow user traffic variability (to avoid statistical multiplexing) and it can be run over PCR or SCR:

. PCR:

$$\mathrm{T} = \frac{1}{\mathrm{PCR}} \qquad \tau = \mathrm{CDVT}|_{\mathrm{PCR}}$$

. SCR:

$$\mathrm{T} = \frac{1}{\mathrm{SCR}} \qquad \tau = \mathrm{IBT} + \mathrm{CDVT}|_{\mathrm{SCR}}$$

QoS

As negotiable parameters, there are several standards avaiable:

. CTD (Cell Transfer Delay): end to end approach; it is the average time between the transmission of the first bit and the reception of the last bit;

. 2-pt CV (Two point Cell Delay Variation);

. CLR (Cell Loss Ratio);

. CER (Cell Error Rate);

. CMR (Cell Misinsertion Rate);

. SECBR (Severely Errored Cell Block Ratio).

First three parameters are the most important. Moreover, some classes have been defined, according to parameters that can be negotiated, to satisfy user services; infact they are related to user's needs, not to network flows. They are:

. class 1: strict (negotiable parameters: CDV, CLR_{0+1});

. class 2: tolerant (negotiable parameters: CLR_{0+1});

. class 3: limited (negotiable parameters: CLR_0);

. class 4: best effort (no negotiable parameters).

This is the minimum set of requirements that operators have to provide for sure, but it is always possible that more classes are defined. As usually, the standard defines only classes, not how users can choose among them or how much a class constraints user's behavior. Notice that the standard does not allow user to negotiate the CTD (the average rate).

Transfer modes

Transfer modes are techniques that specify how ATM network deal with the service; they have been standardized by both ITU-T and ATM forum and, the most important thing to keep in mind, is that they does not define QoS requirements: in practise transfer mode and QoS are completly independent. Transfer modes are chosen per virtual circuit and they can be distinguished through definition of:

. cell flows to which guarantees are provided;

. parameters to characterize flows;

. confermance verification applied to flows;

. adopted control functions.

First two aspects are defined by traffic characterization previously declared.

Most important transfer modes are 5 and they are more suitable with classic QoS needs; they are:

. CBR/DBR: Constant/Deterministic Bit Rate;

. VBR/SBR: Variable/Statistical Bit Rate;

. UBR: Unspecified Bit Rate (best effort traffic);

. ABR: Avaiable Bit Rate;

. ABT: ATM Block Transfer.

Last two modes use RM (Resource Management) cells (one of ATM cell type over which confermance algorithms can run) to control flow cells emission rate.

DBR

DBR is characterized by constant bit rate traffic therefor all cells have the same lenght and inter packet generation time are fixed. Due to these facts, as a parameter, the peak rate is considered to define the transfer mode. Avaiable cells over which PCR is considered are:

. data+OAM+RM;

. data+OAM.

This mode offers always a static bit rate equal to the negotiated peak rate; moreover CAC is performed over B_p, but it is also possible use the equivalent bandwidth: infact QoS parameters are independent so choosing a very tight delay constraint the peak rate is not a valid paramter to make decisions; in this example, a better solution is check the number of admissible calls because the higher is that number the more difficult will be provide a delay guaranteed. This transfer mode is associated to service class 1 (strict) and can be used by both isochronous and fixed bit rate services.

SBR

SBR is charactherized by 3 possible sets:

. SBR1: PCR, SCR, MBS over aggregated flows;

. SBR2: PCR over data cells (0+1), SCR (0), MBS (0); tagging over non conformant cells not admitted;

. SBR3: similar to SBR2, but tagging over non conformant cells admitted.

This transfer mode offers a variable bit rate between PCR and SCR (as discussed for the equivalent bandwidth in CAC) and satisfy source needs instead network needs. Negotiable parameters are typically loss rate and delays and associable classes are both 1, 2 and 3. CAC can be run over the peak rate, the average rate, the equivalent bandwidth or exploiting measurements.

UBR

This transfer mode has been standardized only by ATM forum and the only parameter that is possible declare is the peak rate over aggregated flows: no conformance,no bit rate allocation, no QoS guarantees on delays and no loss probablities are declared because this is the best effort traffic. ITU-T treat it as it is DBR with unspecified class of service. Switches exploits algorithsm and techniques to discarding cells in order to:

. reduce negatives effects due to segmentation:

  . higher packet dropping probability at receiver end;
  . bad use of network resources due to segments belonging to packets that will be dropped by the receiver;

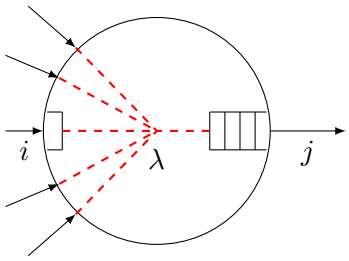
. loss priority in buffers.

Example Consider two cases:

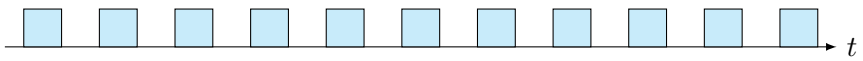
. technology with variable packet size: frame relay;

. technology with fixed packet size: ATM.

In frame relay:



each flow i is a packet. Imagine that only two packets can be stored into the internal buffer: since five packets come, more or less, at the same time, exactly three will be dropped. If, instead, each flow i, is an ATM packet, it is composed by cells:



therefore, in the same situation in which 2 packets (200 cells) can be stored, probably the adopted solution will be store $200/5 = 40$ cells belonging to each flow. After this value, the buffer is full up so packets will be dropped; notice that, in terms of throughput, the two technologies are the same: 3 packets for frame relay and 300 cells for ATM. The essential difference is that in ATM, at the receiver end, the destination can not re-assembly the original packets because tails are lost. Infact, in terms of throughput, ATM is never better that Ethernet or Frame Relay with best effort traffic.

To cope with this facts it is possible:

. use specific algorithms;

. use properly priorities and schedule algorithms (but to schedule, first data have to be stored somewhere).

Cell discarding techniques

. Selective Cell Discarding: this technique drops tails of packets when a cell have been dropped. This is done because, with a cell dropped, the packet will be anyway useless: since only tails are discarded useless traffic is only reduced, not avoided.

. Early Packet Discarding: the entire packet is dropped if, when the first cell comes in there is not enoght space to store it (check the buffer avaiability using thresholds). This is much complex than the previous approach since it is required the knowledge of the packet size at first. Moreover, this is a worst case approach because it is possible that in te near future some more space will be avaiable so more packets are dropped than they should be, but it guaranteeds perfectly that no useless traffic is over the network.

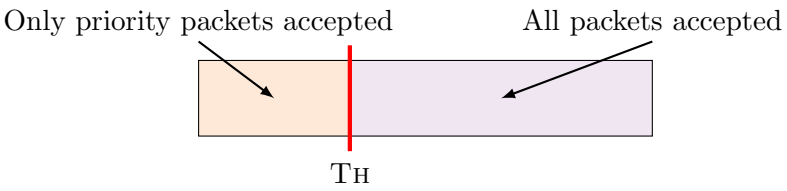
. Use of EFCI bit (congestion indicator): it is set to indicate congestion to highers layer; this approach is not too much used;

. Cell discarding based on priority: when the buffer occupancy is over a given threshold, low priority cells are discarded; it can be implemented in two categories:

  . protective: full separation between high and low priority;

  . non protective.

  For example:



  This approach is not protective because if a low priority packets is accepted and it allow to reach the threashold, an high priority packet that should have come has been penalized. To have complete protection buffer must not be partitioned otherwise some penalization to higher packets sooner or later will happened.

ABR

Time variable bit rate depends only on network avaiability and not on user traffic generation so ABR (Avaiable Bit Rate) try to exploit the avaiable bandwidth allocating a bit rate between the PCR and the MCR. Infact,

using simply the PCR is very difficult saturate the link and it means that the network in always underutilized; instead, using the average rate, sometimes is possible that, for a short period of time, the link is full up, but this is another condition that have to be avoided, otherwise it is difficult to provide QoS. Infact:

. tight QoS parameters $\Longrightarrow$ less network utilization;

. less strict Qos parameters $\Longrightarrow$ high network utilization.

Main ABR objectives are:

. have a full bit rate utilization;

. have resources partitioned in a fair way;

. minimize cell loss.

The third point is due to the fact that ABR exploit principally low priority traffic so congestion can be relevant in term of losses. The minimum bit rate can also be 0 if there are no negotiation, otherwise it is the MCR and it is guaranteed as a CBR source. With this transfer mode, GCRA is run with parameters (T and τ) adaptable in time with network signals.

Source Source traffic is conformant per definition and its behavior is completly defined by standars and to obtain that it is always controlled (speaking about emission bit rate) RM cells are used: infact virtual circuits are bidirectional, therefore crossed nodes are the same going from the source to the destination and from the destination to the source. In few words it is possible resume the source behavior with the following passages:

. at the beginning the trasmission is started with the negotiated bit rate (ICR);

. periodically it inserts RM forward cells in the transmission;

. when it receives a RM backward cell, it adapts the transmission rate with the minimum one reported in the cell;

. if no RM backward cells are received, the source decrease the bit rate until it stops;

. if it has been silent for a long period, when it wakes up it starts transmitting at the ICR.

Node To control source emission rate, nodes can implement three possible solutions:

. EFCI (Explicit Forward Congestion Indicator): the bit, belonging to a data cell, is set to 1 just to signal congestion; this approach is the simple one, but also the less efficient;

. RRM (Relative Rate Marking): with this approach, nodes send RM cell with two bits (CI, NI); in this way it is possible to represent three possible states: increase rate, keep rate and decrease rate; the following table shows it:

CI	NI	MEANING
0	0	Increase
0	1	Keep
1	-	Decrease

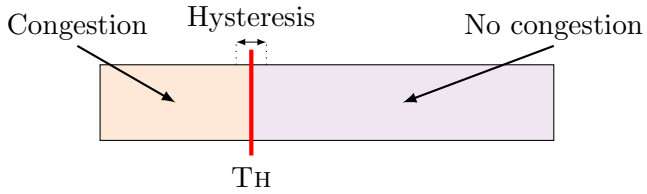
infact a node crossed can only set up to 1 bits, never to 0: it is the default.

. ER (Explicit Rate): with this method, nodes specify within RM cells the proper rate at which the source can transmit.

The three modalities translate for the source to compute rates and use the minimum of them in order to satisfy the bottleneck link.

Threshold When as scheme are adopted EFCI or RRM, nodes exploit congestion control on the base of buffer occupancy and the management of thresholds can be:

. single FIFO queue looking at positions:



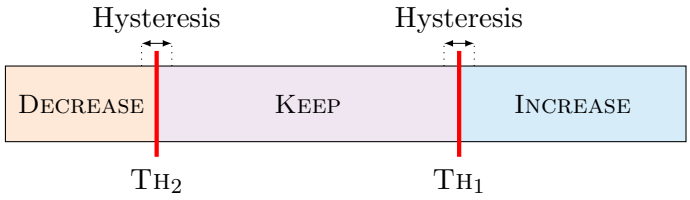
. one FIFO queue for each virtual circuit: it allows to protect high priority packets;

. derivative, looking not at a given instant of time but the behavior on a period (situation in which there is congestion with 100 packets: if the initial state was 10 packets there is a problem, but if the initial state was 300 packets congestion is going over);

. integrative.

Looking at the buffer occupancy is an indirect measure of avaiable bandwidth: infact if the avaiable bandwidth decreases, it means that transfer rates increses, therefore the buffer occupancy decrease. But, in this situation, it is possible that occurs congestion and, as a consequence, transfer rates are reduces so, sooner or later, buffer occupancy increases.

In the case in which RRM is implemented, buffer is divided into three sectors:



The first threshold Th_1 must be positioned close to the empty buffer, in that way the actual used rate will be kept for a longer time avoiding that buffer should fill up sooner; the second threshold Th_2, instead, have to be put as the avaiable space should be sufficient to store packets already sent and packets that the source send before it knows that there is congestion.

ATB

ATB (ATM Transfer Block) has been standardized only by ITU-T and it defines a burst instead of a rate, allowing to negotiate on line QoS block by block. Blocks are groups of cells:

. enclosed by two RM cells;

. preceded by an RM cell.

The service is variable bit rate, bu cells, looking at them independetly, are transmitted at constant bit rate. The bandwidth is allocated block by block with reservation using RM cells; the property of independece is maintain for nodes, that thake decisions again block by block. Due to this fact, a burst reaches the destination only if all nodes accept it, so connection is not guaranteed. Moreover, also the dropping probability is an issue and it is higher the long is the path. Two possible applications for blocks are:

. packets when are segmented;

. frame of videos.

There are two flavous:

. **immediate transmission**: each block is independent and guarantees are provided block by block because they are sent immediatly at a constant bit rate; the node acceptance have not been standardized, but nodes look at buffer occupancy or bit rate;

. **delay transmission**: it is possible renegotiate the block transfer rate with this approach; if it not happends, the initial rate is guaranteed for the entire duration of the connection otherwiese, once the request it is sent, the source have to attend an answer, positive, by the network. ABT have not been su much used in ATM, but nowadays it has been reproposed in optical network.

Discussion about architectures

In this section some possible architectures are considerd for all ATM modes; the discussion will be about:

. queuing structure;

. scheduler.

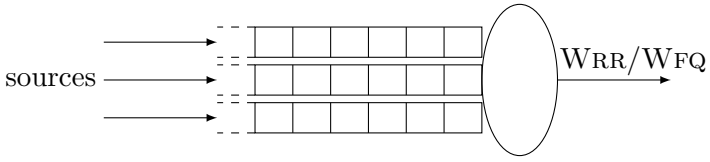
First each transfer mode will be considered independently, then all together.

Solution

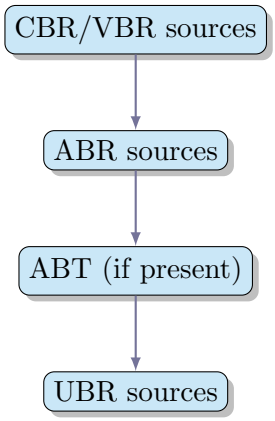
Two main queuing structures are avaiable:

. one single queue served in FIFO order (mode 1):

. one queue for each flow served in WRR or WFQ (mode 2):



Transfer modes are analized according to the level priority:

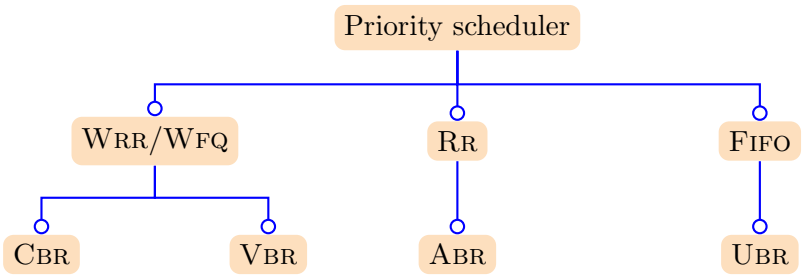


Infact, usually CBR and VBR are exploited for sure, ABR exploits the residual avaiable bandwidth and best effort traffic, UBR, is exploited if possible.

- **CBR**: both modes can be exploited; mode 1, guarantees for sure bandwidth but not delays: this is due to non conformant flows (traffic shape may change). The second mode is more robust and guarantees delays since they are independent for each queue. Of course all considerations depend on what kind of service have to be provided: for example if, for mode 1, delays are not a tight constraint, it is possible to bound them under-utilizing network resources (at 60% for example); using this method the equivalent bandwith have to be computed for entrance flows.
- **VBR**: all considerations done before are still valid. The main difference is that, for acceptance, the only method that is possible use is the equivalent bandwidth:
  - if as parameter is considered the average rate, no guaranteed are provided a part from the istantaneous rate;
  - if as parameter is considered the peak rate, the bandwidth is well guaranteed, but not delays and, as a usual consequence when this parameter is chosen, the network resources are under-utilized;
  - the equivalent bandwidth is the only parameter that allows to allocate exactly only necessary resources for the call.
- **UBR**: since it is the best effort traffic, it is sufficient mode 1.
- **ABR**: since this approach works checking the buffer for EFCI and RRM, it is better have only one buffer, therefore the better and easier

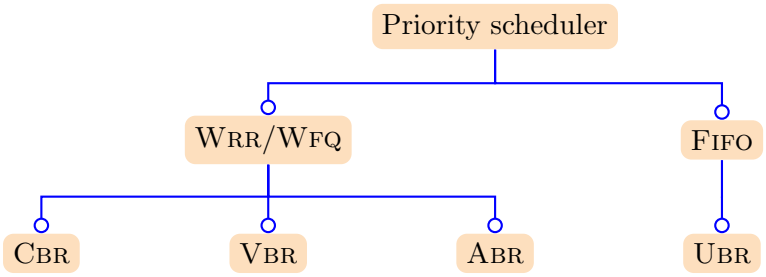
method is mode 1; using ER, it is possible also implemented mode 2, but if the ERICA algorithm is implemented, again, the better choice is mode 1. Instead, if the goal is protect flows, the only mode that is possible use is mode 2 because it is the only one that uses WRR or WFQ: notice that it happends also if the traffic is not policed, infact, from user's point of view is equal if policing is done at beginning or in networks nodes with scheduling.

. **All modes together**: in this case the following scheme resume the adopted behavior.



Scheduling is hierarchical because the first level is just needed to chose among flows, while the priority scheduler gives more relevance to CBR and VBR that actually are the two with higher priority; if there is some residual bandwidth (it means that CBR and VBR buffers are locally empty) it exploits ABR and, only at last, if it is possible the best effort traffic is transmitted.

This scheme works only if ABR has no constraints over the minimum rate MCR otherwise the scheme have to be changed in this way:



ABR without MCR is similar to UBR so it is also possible avoid CAC, but if MCR is negotiated CAC is mandatory.

Notice that another solution can be implemented: trunking, the static resource partitioning.