Traffic Class Expediting for ATM LAN Emulation

Bob Klessig
Mick Seaman
3Com Corporation

At present LAN Emulation hides the benefits of ATM QoS from users' applications. This note discusses support of 802.1p by ATM QoS via LUNI v2.0. It describes where and how frames with different user priorities are mapped to traffic class queues, and discusses the alternatives and issues involved in relating frame queues to ATM VCCs.

This note proposes mapping each priority into an ABR VCC where the ABR parameters are different for different priorities. Analysis indicates that this approach should be easy to configure and will automatically allocate bandwidth on the basis of the traffic's user priority.

ATM Edge Devices

Consider an ATM edge device, i.e. a LAN to LAN Emulation Bridge, with 'legacy' LAN ports and one or more ATM ports. For frames destined to a legacy LAN port, the edge device should map the frames to multiple traffic class queues, and service those queues as specified by 802.1p [1][1]. For frames destined to an ATM port, the edge device should also maintain and service multiple queues in the same way.

ATM attached End Systems

If an ATM attached end system using LANE has an 802.1p capable driver, then each frame sent by the end system will have a user_priority (possibly the default, priority 0). In this case, the host behavior should mirror that of the edge device: maintaining multiple queues, mapping frames to queues, and servicing those queues.

ATM Transmission

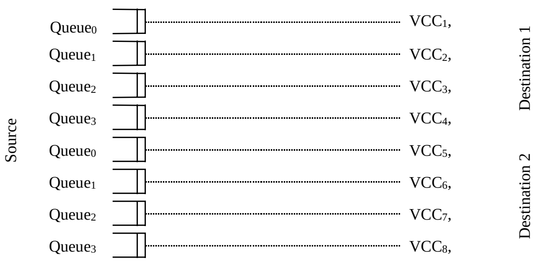
Both in the case of an ATM edge device and an attached end system there remains the question of how to relate VCCs to the frame queues for transmission over the ATM interface. The simplest approach is to use a single VCC for (the cells of) all frames destined for a given ATM address, independent of the frame queue. The 802.1p priorities are maintained within the VCC. However, at ATM switches where cells from different VCCs are multiplexed on outgoing links, cells from frames with lower priorities can delay (or cause to be discarded) cells from higher priority frames. This is because the ATM switch can only distinguish VCCs and not frame priorities in processing a cell. Effectively, the priorities become diluted.[2]

To reduce the dilution of priorities described above, a more complex approach can be used. For each destination ATM address, multiple frame queues are established. A different VCC is used to transmit cells from each frame queue and each VCC has a different QoS. Example 1 shows four frame queues and two destinations. The result is 8 VCCs and 4 different values of QoS. If we

[1] Clause 7 specifies queue operation. 7.7.4 specifies the default queue servicing.

[2] The degree of dilution will depend on the queuing technique used in the ATM switch. If per VCC queues are used the dilution effect will probably be modest. If all of the VCCs have the same QoS parameters and if the switch only maintains a single queue for each QoS, then the dilution could be significant.

can choose the values of the QoS appropriately, an ATM switch can distinguish between cells of different IEEE 802.1p priority.



Example 1 - Multiple VCCs with differing QoS

There are three challenges in choosing the values of QoS:

1. There is no way to choose QoS values such that cells from a frame with higher priority will always have priority over cells from a frame with lower priority.
2. Even if pure priority cannot be maintained, it is still desirable to specify the values of QoS such that there is an ordering. In the example above the ordering should be $QoS_0 < QoS_1 < QoS_2 < QoS_3$. However, since QoS is specified by a vector of parameters, there is no obvious ordering.
3. In order to specify a QoS when setting up a VCC, it is necessary to specify a cell rate that the traffic will not exceed. Unfortunately, the user_priority quite deliberately does not give any information about how much bandwidth is required. Algorithms that predict future bandwidth needs by observing recent traffic levels could be invented but these would probably lead to very complex implementations.[3] Alternatively the problem could be relegated to network management but this would imply a heavy burden on the network manager to select and monitor many bandwidth levels.

Available Bit Rate Mapping

We propose the use of ABR [3] to address challenges 2 and 3 above. The key parameters that define an ABR VCC are:

- Peak Cell Rate (PCR): In this mapping approach, all ABR VCCs on a given ATM interface should have the same value of PCR. In the absence of other traffic, PCR should equal the cell rate of the physical link. If other kinds of traffic, such as CBR, are being carried across the interface, then PCR can be appropriately reduced.
- Minimum Cell Rate (MCR): In this mapping approach, all ABR VCCs on a given ATM interface should have the same value of MCR. The value should be a small percentage of PCR. In the analysis that follows, we have assumed

$$MCR = .02 \times PCR$$

- Rate Decrease Factor (RDF): The value is set based on the QoS that is desired. Recall that the Actual Cell Rate (ACR) is reduced by this factor (but not below MCR) when congestion is detected in the ATM network. Stating this in the form of an equation we obtain

$$ACR_{new} = \max\left(MCR, RDF \times ACR_{old}\right)$$

- Rate Increase Factor (RIF): The value is set based on the QoS that is desired. Recall that the ACR is increased by adding $RIF \times PCR$ (but not above PCR) when congestion is absent. Stating this in the form of an equation we obtain

$$ACR_{new} = \min\left(PCR, ACR_{old} + RIF \times PCR\right)$$

Given that the PCR and MCR are the same for all VCCs, we can represent the QoS of a VCC by a vector $QoS = \langle RDF, RIF \rangle$. Intuitively we can see that QoS is better for larger values of RDF and RIF. This allows a simple (partial?) ordering to be defined for QoS.

[3] Such efforts would be better directed at the stock market.

For $QoS_1 = \langle RDF_1, RIF_1 \rangle$ and $QoS_2 = \langle RDF_2, RIF_2 \rangle$, $QoS_1 < QoS_2$ if and only if $RDF_1 < RDF_2$ and $RIF_1 < RIF_2$.

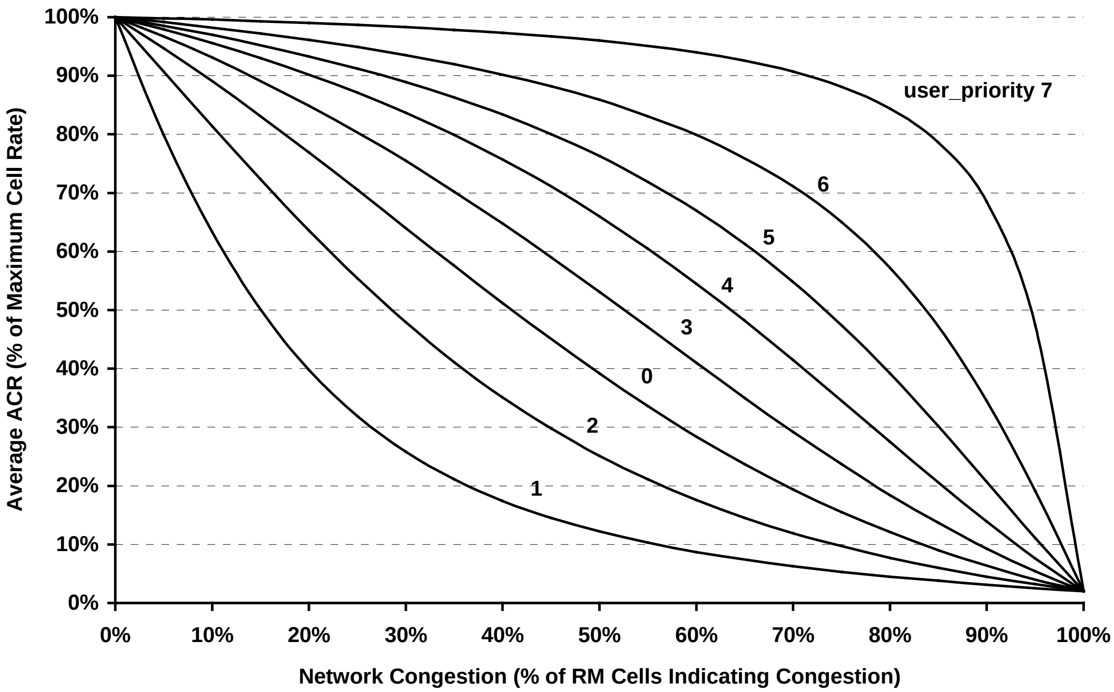
We see that we have met challenges 2 and 3 above. It is straight forward to pick RDF and RIF for each VCC such that the higher priority frame queues are serviced by better QoS VCCs. Furthermore, there is no need to predict the bandwidth for each IEEE 802.1p priority for a given destination. Choosing PCR and MCR is straightforward.

When there is no congestion in the network, traffic is free to flow through all frame queues. Each priority is free to use the bandwidth it needs. If traffic exceeds the bandwidth of the physical link, then the frame priority queue service rules will ensure that the higher priority frames get the bandwidth they need at the expense of the lower priority frames. When congestion develops in the ATM network or at the destination ATM terminal, the backward RM cells[4] will indicate congestion. The ACR for the highest priority VCC will be reduced modestly while the ACR for the lowest priority VCC will be reduced aggressively.

When a backward RM cell arrives indicating no congestion, the ACR for the highest priority VCC will increase aggressively while the ACR for the lowest priority VCC will increase modestly. Thus, in the face of network congestion, the bandwidth is taken away from the lowest priority VCC first. As more congestion develops, bandwidth limits start working their way up the priority list.

The remaining question is how to set the RDF and RIF values to attain the proper amount of preference for higher priorities over lower priorities. As an aid to understanding this question and to get a more quantitative feel for the behavior, a model was developed (details below). The model calculates the average ACR for each priority as a function of the percentage of backward RM cells indicating congestion. A device with 8 frame queues is modeled. MCR is set at 2% of PCR. The results shown below confirm that bandwidth is "taken away" from the low priority traffic when congestion occurs.

[4] For purposes of this discussion we assume that explicit rate capability is not present.



Of course the model makes many strong assumptions and is probably not a sufficient basis for choosing the parameters in a real network. What it tells us is that the mapping approach probably provides a sufficiently rich set of controls to allow a network manager to achieve meaningful discrimination of 802.1p priorities within the ATM network.

Enhanced Queue Servicing

The 802.1p rules for servicing transmission queues could be taken to imply that one frame should be completely transmitted before starting another frame transmission. When network congestion occurs this leads to less than full use of the ATM bandwidth, even when several frames are waiting to be sent. Suppose that congestion has led the ACR for the VCC serving $Queue_7$ to have ACR = 80% of the line rate. This means that during the transmission of a frame from $Queue_n$, 20% of the cells on the ATM link would be empty. This would be the case even if there was a frame in $Queue_{n-1}$ waiting to be transmitted on a different VCC.

An interesting variation on this effect could occur if the VCCs to a particular destination follow different paths through the ATM network. This can happen since the VCCs can be set up at different times. If the VCC for $Queue_n$ passes through a part of the network with high congestion, it could find its ACR substantially reduced. At the same time, the VCC for $Queue_{n-1}$ may not see any congestion and it has a full value for ACR. The effect would be that frames in $Queue_{n-1}$ would be held up by the higher priority frames even though there is plenty of bandwidth available for $Queue_{n-1}$.

To avoid this phenomenon, a cell oriented queue service algorithm can be used. Suppose a cell is ready for transmission on the ATM link. If sending that cell on the VCC serving $Queue_n$ will cause the ACR to be exceeded, we can say that $Queue_n$ is *paused*. With this definition we can clarify the queue servicing rules as follows:

For each cell transmit opportunity on the ATM link, select a cell from the highest priority frame queue that contains a (partial frame) and is not paused.

This approach will improve the efficiency of the use of the ATM link. It will increase the probability of a lower priority frame being sent ahead of a higher priority frame. For example, a frame that is 10 cells long in $Queue_n$ could be passed by a frame that is 2 cells long in $Queue_{n-1}$.

References

[1] IEEE Project 802, *Information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Common specifications — Part 3: Media Access Control (MAC) Bridges: Revision (Incorporating IEEE P802.1p: Traffic Class Expediting and Dynamic Multicast Filtering)*, ISO/IEC Final CD 15802-3, IEEE P802.1D/D15, November 24, 1997.

[2] ATM Forum, *ATM LAN Emulation Version 2 - LUNI Specification*, af-lane-0084.000, July 1997.

[3] ATM Forum, *Traffic Management Specification Version 4.0*, af-tm-0056.000, April 1996.

[4] IEEE Project 802, *Draft Standard P802.1Q/D9 IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks*, P802.1Q/D9, February 20, 1998.

Markov Model Details

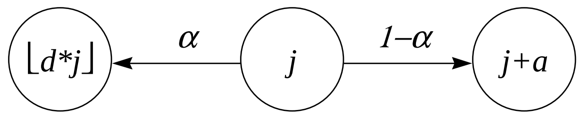
We model the QoS mapping as a Markov Process. The states of the model are numbered 1 through 50 with 50 representing 100% of the bandwidth of the ATM link (in cells per second) and 1 representing 2% of the bandwidth. For simplicity, we take 2% as the ABR Minimum cell rate and 100% as the ABR Maximum cell rate. A state transition occurs each time a Resource Management cell is received that either indicates congestion (implying a multiplicative decrease in the cell rate) or no congestion (implying an additive increase in the cell rate). The key simplifying assumption is that the probability of receiving a congestion indication is independent of the state.

Table 1 shows the basic parameters of the model.

α	probability of congestion indication
j	= ACR/.02
a	RIF × 100% (> 0)
d	RDF (< 1)
j_{min}	minimum ACR index (= 1)
j_{max}	maximum ACR index (= 50)
$\pi_j(n)$	probability that cell rate is j at epoch n

Table 1. Model parameters

The state transitions from state j are illustrated below. The floor function is used in the case of cell rate decrease to insure that there is always a decrease in state. Note that an RM cell can also indicate a “no change” in cell rate. This would have the effect of the process remaining in the state. We can ignore this case because it would not change the steady state probabilities. When j is close to either j_{min} or j_{max} , the transitions have to be adjusted to reflect that there are bounds on the value of j.



State Transition for *j* far from j_{min} *and* j_{max}

We now derive the equations for the state probabilities. First we define some sets of states.

$$M(j,d) \equiv \{m' | \lfloor dm' \rfloor = j, 1 \le m' \le j_{\max}, m' = integer\}$$

Equation 1

$M(j,d)$ is the set of states from which j is reachable via a decrease in the cell rate.

$$K(d) \equiv \{k' | \lfloor dk' \rfloor \le j_{\min}, 1 \le k' \le j_{\max}, k' = integer\}$$

Equation 2

$K(d)$ is the set of states from which j_{min} is reachable via a decrease in the cell rate. Notice that $M(j_{\min}, d) \subseteq K(d)$ because any attempt to lower the cell rate below j_{min} results in the new cell rate being exactly j_{min}. Also notice that j_{min} can only be reached via a cell rate decrease.

$$L(a) \equiv \{l' | l' \ge \max(j_{\max} - a, j_{\min}), l' = integer\}$$

Equation 3

$L(a)$ is the set of states from which j_{max} can be reached via a cell rate increase. Notice that j_{max} can only be reached via a cell rate increase and there can be several such states because any attempt to increase the cell rate above j_{max} results in the new cell rate being exactly j_{max}.

Then for $j_{\min} < j < j_{\max}$,

$$\pi_j(n) = (1-\alpha)\pi_{j-a}(n-1) + \alpha \sum_{i \in M(j,d)} \pi_i(n-1)$$

when $j - a \ge j_{\min}$,

Equation 4

$$\pi_j(n) = \alpha \sum_{i \in M(j,d)} \pi_i(n-1)$$

when $j - a < j_{\min}$.

Equation 5

and

$$\pi_{j_{\min}}(n) = \alpha \sum_{i \in K(d)} \pi_i(n-1),$$

Equation 6

$$\pi_{j_{\max}}(n) = (1-\alpha) \sum_{i \in L(a)} \pi_i(n-1).$$

Equation 7

Because of the way the state transitions occur, if $\sum_{j=j_{\min}}^{j_{\max}} \pi_j(0) = 1$, then

$$\sum_{j=j_{\min}}^{j_{\max}} \pi_j(n) = 1 \qquad \text{for all } n.$$

Equation 8

If we denote $\hat{\pi}_j = \lim_{n\to\infty} \pi_j(n)$, then plugging $\hat{\pi}_j$ into the above equations allows us to solve for $\hat{\pi}_j$. From there the mean ACR in % is given by

$$mean(a,d,\alpha) = \left(\frac{1}{50}\right) \sum_{j=1}^{50} j\hat{\pi}_j$$

Equation 9

What remains is the choice of a and d for each priority. For high priority traffic, d should be close to 1 while a should be large. For purposes of example, the parameters shown in Table 2 are used. The results of the calculations for the mean using these parameters are displayed in the graph above.

user_priority	Frame Priority	RIF	*a*	*d* = RDF
7	7	.40	40	.96
6	6	.35	35	.84
5	5	.30	30	.72
4	4	.25	25	.60
3	3	.20	20	.48
0	2	.15	15	.36
2	1	.10	10	.24
1	0	.05	5	.12

Table 2. Parameters for Each Priority