



**HAL**  
open science

## Classification of service for entertainment applications in vehicles

Jean-Philippe Georges, Eric Rondeau, Thierry Divoux

► **To cite this version:**

Jean-Philippe Georges, Eric Rondeau, Thierry Divoux. Classification of service for entertainment applications in vehicles. International Conference on Internet and Web Applications and Services, Feb 2006, France. hal-00101542

**HAL Id: hal-00101542**

**<https://hal.science/hal-00101542v1>**

Submitted on 29 Sep 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Classification of service for entertainment applications in vehicles

Jean-Philippe Georges, Eric Rondeau, Thierry Divoux  
Université Henri Poincaré Nancy 1

Centre de Recherche en Automatique de Nancy (UMR 7039 CNRS-INPL-UHP)  
Faculté des Sciences, BP 239, F-54600 Vandœuvre lès Nancy cedex, France  
<firstname.surname>@cran.uhp-nancy.fr

## Abstract

*Generally, the real-time applications exchange information on dedicated network and the other traffic is supported by another communication system. The differentiation of these two networks enables to easily guarantee the level of Quality of Service required by the application. The objective of this paper is to study Ethernet network architectures which transport all kinds of information. The mechanism of Classification of Service are modelled by using Network Calculus to adjust the service offered by the network with the information timed requirements. The study analyzes two kinds of scheduling policies and could be used in the framework of entertainment applications in vehicles.*

## 1. Introduction

Vehicles use to interconnect their equipments specifics networks such as Controller Area Network (CAN), Vehicle Area Network (VAN) for cars and for example ARINC protocol in avionics systems. Generally, the real-time applications exchange information on dedicated network and the other traffic is supported by another communication system. It enables to easily guarantee the level of Quality of Service required by the control of the vehicles (brake, ESP...). The new trends which we can observe are to implement only one network to transport all kinds of information. For example, Renault Company has implemented only one CAN network to support all kinds of information exchanges in the Vel-Satis model and EADS/Airbus have chosen switched Ethernet technology for the communications in the new Airbus 380. In the same time, the vehicle constructors propose more and more embedded entertainment applications which require more and more network bandwidth. In this context, Thales Company studies the implementation of entertainment applications based on WIFI communications in the future Boeing 787 Dreamliner. The goal is to propose to the passenger internet connections, video on demand, etc.

The objective of this paper is to propose a general method enabling to analyze the behaviour of Ethernet network architecture used as a shared resource for both real-time and entertainment traffic embedded in vehicles (figure 1). At this time, the use of a unique network to transport all kinds on information seems to be unrealistic. Nevertheless it is important to anticipate such communication organization which has an economical impact since it enables to reduce considerably the cabling costs, and then to reduce the wire weight on the vehicle. The first step of this paper is to study the means to be able to differentiate the services offered by the network to manage all kinds of traffic. One of the mechanisms proposed in Ethernet standards is to implement the Class of Service (CoS) concept. The CoS enables to associate the time constraints of the application with the service offered by the network.

Due to its non-deterministic MAC protocol [1], Ethernet has just been used for non-time-sensitive applications. Nevertheless, due to its performances (transmission rates, interoperability, flexibility, standardization), it is interesting to use Ethernet in time-critical applications. Different works have also been achieved to resolve the random CSMA/CD medium access arbitration in modifying the MAC algorithm [10] or in adding an upper-level arbitration protocol [12][8]. In fact, collisions could be eliminated by using the switch technology [6]. Switches break up collision domains into small groups of devices, effectively reducing the number of collisions. Furthermore, with micro-segmentation, each device is isolated in its own segment in full-duplex mode and has the entire port throughput for its own use. But in these conditions, the collision problem is shifted to congestion inside switches. So, all these considerations do not prove that all the frames are effectively received under a predefined temporal bound specified by the applications. The reason is that at the layer 2 of the OSI model, no Quality of Service (QoS) can be managed, but only the *Classification of Service* (CoS) is available. The QoS mechanisms enable to guarantee some performances regarding explicit application requirements. For that, some resources such as channels

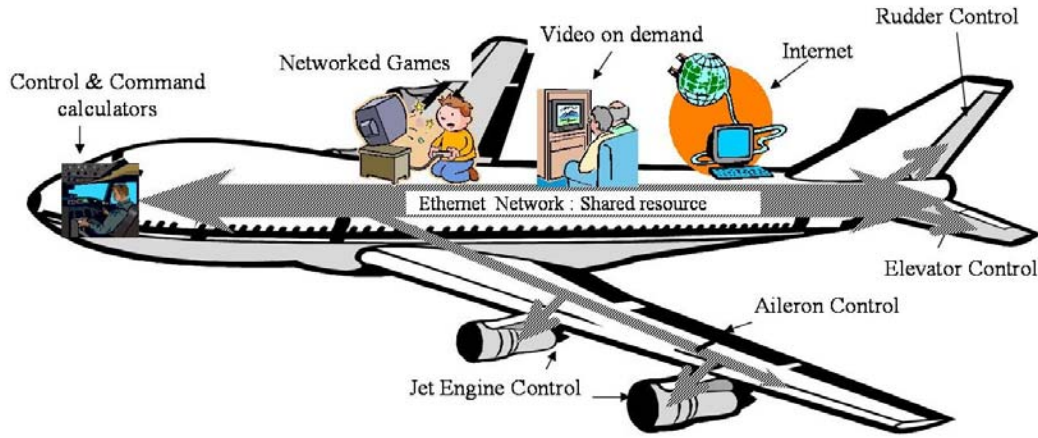


Figure 1. A shared network in vehicles

and memory are reserved. In opposition, the CoS mechanisms only provide a better service for critical applications. But they do not explicitly guarantee that specific needs will be satisfied. That is why their efficiency has to be verified by using analytical evaluation techniques. In this paper, the Network Calculus theory [9] will be applied to be sure that the communication network will meet the application constraints.

## 2. Analysis of the traffic characteristics

At first, the traffic handled by time-critical applications could be classified from different points of view. Several kinds of messages could be identified regarding their periodicity and their temporal deadlines:

- the control traffic is usually periodic and is hardly time-constrained. It depends on the time cycles defined by the control devices such as calculators. Volume and frequency are well known.
- alarms are sent asynchronously, but have also to be received under a predefined bounded time.
- the traffic for entertainment and comfort applications (video, networking games, Internet access) are also asynchronous, but do not have to respect so strict hard time constraints. This traffic is defined by the users, so that it is generally unknown.

The priority level of the messages will be consequently determined by the real-time requirements of the information. Alarms present the hardest time constraints, so that alarms frames will be tagged with the highest priority. Then the control traffic has to be served, and finally the asynchronous traffic has the lowest priority.

Then, since the objective is to guarantee some deterministic performances of the network, the incoming traffic has to be bounded, and consequently, it has to be constrained. Generally, constraints are specified by a regulation method, like the leaky bucket controller concept [9]. In this scheme, data will arrive at the leaky bucket rate only if the level of the bucket is inferior to the maximum bucket size. Now, lets  $R(t)$  to represent the instantaneous rate of the stream,  $\sigma$  to be the maximum amount of traffic that can arrive in a burst and  $\rho$  to be an upper bound on the long-term average rate of the traffic. A similar expression of the leaky bucket constraint is the burstiness constraint defined in [2]. The traffic modelling is done by a  $(\sigma, \rho)$  envelop called  $b(t)$ , such as  $b(t) = \sigma + \rho t$ . If  $R(t)$  represents the instantaneous rate of data, the burstiness constraint is written  $R \sim b$  as defined  $\forall x, y; y \geq x, x \geq 0$ , then

$$R \sim b \Leftrightarrow \int_x^y R(t) dt \leq \sigma + \rho(y - x)$$

The issue is now to identify for a given traffic the value of  $\sigma$  and  $\rho$  such that the arrival of traffic will be bounded. Moreover, a supplementary constraint will be integrated in the traffic model: on a network, we know that the data arrival rate is limited by the capacity of the link which will be noted  $C$ . It corresponds to a stability constraint, such that  $b(x) \leq Cx$ . Consequently, the arrival curve is:

$$b(t) = \min \{Ct, \sigma + \rho t\} \quad (1)$$

## 3. Classification of service in switched Ethernet networks

The native Ethernet does not implement any priority mechanism. Several solutions have been proposed but non standardized such as adapting the interframe gap (smaller

for high priority frames), modifying the *Binary Exponential Backoff* algorithm (the waiting time is not randomly calculated, but in relation with the priority), or using a variable length of the preamble (smaller for high priorities). Another approach consists in using a Time Division Multiple Access method over the native CSMA/CD protocol: pre-allocated time-slots are defined for the transmission of time-critical data.

Nevertheless, the evolution of Ethernet to segmented architectures and the definition of the *Virtual Local Area Networks* (VLAN) have led to the emergence of a new standards set (802.1D/p, 802.1Q) in which new encapsulation fields are added to the classical frame [7]. One of these fields is specified in order to support 8 priority levels associated to 8 types of applications (voice, video, best effort, etc.). The number of Classes of Service may be different to the number of priority levels, and also different for each port. That is why the standard also recommends a mapping between classes, priority and ports queues.

The next point is the scheduling policy that will be used to forward the frames at the output port regarding their dedicated priorities. [7, section 8.6.6] defines two items.

- for a given supported value of traffic class, frames are selected from the corresponding queue for transmission only if all queues corresponding to numerically higher values of traffic class supported by the port are empty at the time of selection;
- for a given queue, the order of which frames are selected shall maintain the incoming ordering.

That is to say that the scheduling policy defined is the *Strict Priority (SP)* algorithm and that the policy must be FIFO for a given queue. But the standard enables to implement other algorithms. The main drawback of the SP algorithm is that it can lead to the impossibility for the lowest priority queues to be forwarded. To resolve it, CoS switches implement a supplementary policy : the *Weighted Fair Queueing (WFQ)*. In the Fair Queueing algorithms, the service offered to the high priority queues is moderated as following. A weight is associated to each queue. Then the scheduler gives to each queue (from the highest priority to the lowest) a bandwidth determined by its associated weight.

The Weighted Fair Queueing, initially proposed in [3], is also known as the Packetized Generalized Processor Sharing (PGPS). It is based on the conceptual algorithm called the Generalized Processor Sharing (GPS) [11]. However practical implementations of WFQ in today's switch products are based on a simplified Weighted Round Robin.

In a round robin policy, packets are pushed in queues according to their priority level. Then, the server pools the different queues according to a cyclic sequence (using a pre computed order defined by the queues priorities) in an attempt to serve one packet for each non empty queue. Even

if this algorithm respects the fairness quality, no flexibility is integrated. Moreover, the fairness can be damaged with variable packet lengths. To improve the lack of flexibility of a simple round robin policy, the Weighted Round Robin (WRR) associates a weight  $w_i$  on each flow  $i$ . Now, the WRR server will attempt to serve a flow  $i$  with a rate of  $\frac{w_i}{\sum_j w_j}$  before looking for the following queue. Comparing to PGPS, delays could be more important since if the system is heavily loaded and a frame just misses its slot, it will have to wait its next slot, i.e. a cycle.

In this paper, we will study a WFQ policy based on a per-priority queuing and a weighted round robin scheduling. This implementation is typical of switch products, like the *Cisco Catalyst 2950*. In the next part, the service offered by the SP and WFQ scheduling are compared.

## 4. Analysis of SP and WFQ policies

In this section, we consider a node with three inputs. The capacity of the input ports is fixed at  $C_{in}$  b/s and the capacity of the output at  $C$  b/s, such that  $C_{in} \geq C$ . Each flow is  $(\sigma_i, \rho_i, C_{in})$  upper-constrained with  $\sum_j \rho_j < C$  and a weight  $\phi_i$  is given to each flow. We will suppose that  $\phi_i > \phi_{i+1}$ . Concerning the departure curve of a flow, we could already write that  $R_i^*(t) \leq Ct$ , i.e. the amount of data forwarded for a flow  $i$  is inferior to the capacity of the output port.

### 4.1. Strict Priority

In the strict priority policy, none guarantee is offered to one flow. The selection order will simply depend on the priority (weight) order. Also, we have to distinguish the service curve offered to each flow. The maximum length of packets belonging to a flow  $i$  will be noted  $L_{i,max}$ . In [5], the service curve offered to one flow corresponds to a rate latency service curve :

$$\beta(t) = r(t - T)^+$$

$r$  corresponds to the service rate and  $T$  to the latency introduced by the scheduler. They are defined for :

- the high priority:

$$T = \max\{L_{2,max}, L_{3,max}\} / C, \quad r = C$$

- the medium priority:

$$T = \frac{\sigma_1}{C - \rho_1} + \frac{L_{3,max}}{C}, \quad r = C - \rho_1$$

- the poor priority

$$T = \frac{\sigma_1 + \sigma_2}{C - \rho_1 - \rho_2}, \quad r = C - \rho_1 - \rho_2$$

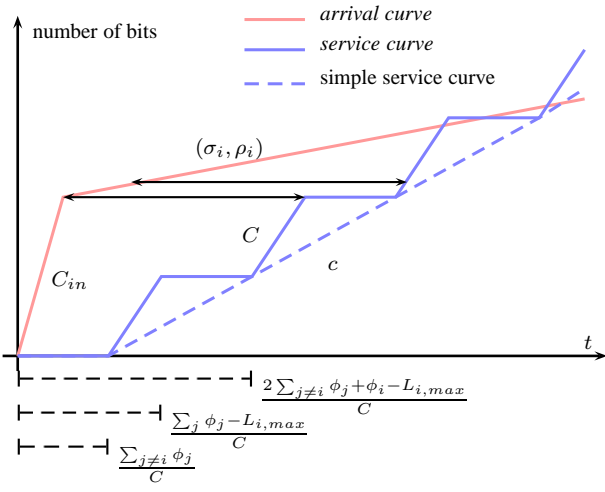


Figure 2. Service curve weighted round robin.

#### 4.2. Fair queueing and Weighted round robin

The fairness principle imposes that the service offered to one flow does not depend on the  $(\sigma, \rho)$  properties of the other flows. Since we want to offer a better service to time constrained traffic, weights (noted  $\phi_i$  for flow with the priority  $i$ ) will enable to limit the service according to the priority level of a packet.

In a round robin scheme, for every non-empty queue encountered, the node will try to serve up to  $w_i$  packets before moving to the next queue in the round. Since the fairness of this solution is not robust to variable-length packets, it will be stipulated that no more  $\phi_i$  units of a flow's traffic are served at each time the flow is polled. The maximum length of a round is consequently equal to  $\frac{\sum_j \phi_j}{C}$  and the time for the packet of the flow  $i$  to be forwarded inside a round is bounded by  $\frac{\phi_i}{C}$ . Moreover, the fairness principle imposes that the server does not provide to a given flow more than  $\phi_i$  units of data. Since the forwarding of packet to the network cannot be pre-empted, the server might refuse to forward a packet if its length is superior to the quantity of units remaining. So that even if the queue of a given flow will never be empty, the units of data offered to the flow per round will be limited in the worst case to  $\phi_i - L_{i,max}$ . At all, in opposition to the strict priority, the service offered to one flow only depends on the weights of the flows and on its properties : it respects the fairness queuing.

The definition of the minimal service curve offered to the flow will use the properties given above, but has also to consider that the worst case will append when a packet just misses its slot in the current round, so that it will have to wait its next slot (at the next round). In the worst case, it will be supposed that other flows are waiting, so the packet will have to wait up to  $\frac{\sum_{j \neq i} \phi_j}{C}$ . The figure 2 shows the

obtained service curve and the equation (2) its formulation.

$$\beta(t) = C \left( t - \left\lceil \frac{t}{\frac{\sum_j \phi_j - L_{i,max}}{C}} \right\rceil \frac{\sum_{j \neq i} \phi_j}{C} \right)^+ \quad (2)$$

The figure 2 shows the arrival curve (given by equation (1)) and the service curve obtained previously. This curve entitled *service curve* well takes into account the cycle introduced by WFQ for a given flow. In fact, this curve highlights the fairness of WFQ since the kind of service curve will be the same for all flows. The difference between two flows with different priorities will just appear in the numerical application (the value of  $\phi_i$  will change according to the priority of the flow  $i$ ).

Since the equation (2) can be difficult to analyze, it is noticed that another service curve can be proposed.

$$\beta_{r,T}(t) = r(t - T)^+ \quad (3)$$

$$r = C \frac{\phi_i - L_{i,max}}{\sum_j \phi_j - L_{i,max}}, \quad T = \frac{\sum_{j \neq i} \phi_j}{C}$$

This new service curve is added on the figure 2 (curve entitled *simple service curve*). We can see that this new service curve is more pessimistic but also easier than the previous. The two equations (2) and (3) show that the service curve offered in the weighted round robin scheduling to a given flow only depends on the weights, so that it is fairness and flexible (as WFQ is fairness, the general service offered to one flow is always the same, only values will be different and take into account the priority level of the flow). By simplicity, only the equation (3) will be considered in the following. Now, we look at the delay.

#### 4.3. Bounded delays for SP and WFQ

As it is shown on the figure 2, [9] has observed that the delay corresponds to the horizontal deviation between the arrival and the service curve. In [5], we have demonstrated that the delay is bounded by :

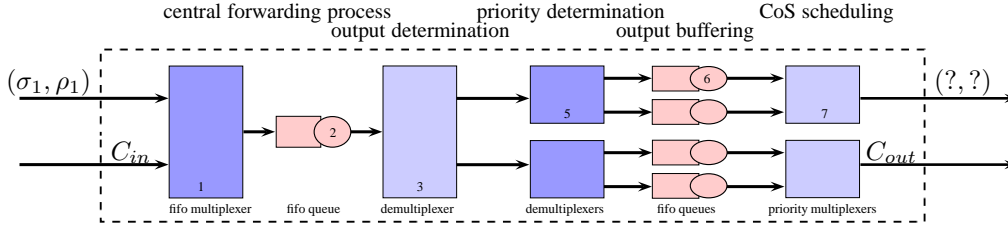
$$\overline{D}_i = (T - \tau_i) + \frac{\sigma_i + \rho_i \tau_i}{r} \quad (4)$$

Using the equation (4), it is now possible to compare delays provided by strict priority and weighted round robin. First, consider the flow 1 of our example. We have:

$$\overline{D}_{1,SP} = \left( \frac{\max(L_{2,max} + L_{3,max})}{C} - \tau_1 \right) + \frac{\sigma_1 + \rho_1 \tau_1}{C}$$

$$\overline{D}_{1,WR} = \left( \frac{\phi_2 + \phi_3}{C} - \tau_1 \right) + \frac{\sigma_1 + \rho_1 \tau_1}{C \frac{\phi_1 - L_{1,max}}{\phi_1 + \phi_2 + \phi_3 - L_{1,max}}}$$

It is interesting to note here that if  $\phi_2 \rightarrow 0$  and  $\phi_3 \rightarrow 0$ , the weighted round robin will be very closed to the strict policy.



**Figure 3. The prioritisation mechanism modifies the output buffering at output ports**

But if we consider now the flow 3, we have:

$$\overline{D_{3,SP}} = \left( \frac{\sigma_1 + \sigma_2}{C - \rho_1 - \rho_2} - \tau_3 \right) + \frac{\sigma_3 + \rho_3 \tau_3}{C - \rho_1 - \rho_2}$$

$$\overline{D_{3,WRR}} = \left( \frac{\phi_1 + \phi_2}{C} - \tau_3 \right) + \frac{\sigma_3 + \rho_3 \tau_3}{C \frac{\phi_3 - L_{3,max}}{\phi_1 + \phi_2 + \phi_3 - L_{3,max}}}$$

That shows that if  $\rho_1 + \rho_2 \rightarrow C$ , the forwarding rate offered to the packet with a lower priority will tend to zero in the Strict Priority scheduling. This problem does not appear in the weighted round robin policy, since the forwarding rate will only depends on the weights' values. One of the next issues will be also the determination of the weights  $\phi_j$  values.

## 5. Switch service modelling

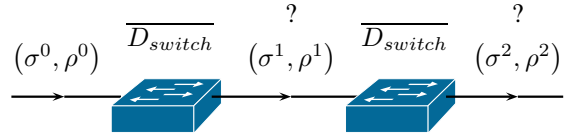
One of the central issues of the performance evaluation of a switched network is the characterisation of the service offered by a switch. The figure 3 shows the switch modelling (developed in [4]) that will be used in this paper. It is based on service curve and take into account the different mechanisms implemented inside the switches, such as the forwarding process (switching of a frame to one output port) or the priority management (802.1p). The figure 3 represents a two-ports switch model which is able to manage the frames forwarding with two levels of priorities.

The implementation of priorities inside a switch mainly consists in adding for each switch output port as many buffers as priorities. These buffers are modelled by queues (component 6 on the figure 3). The switching operation on the FIFO queues belonging to a same switch output port is achieved by a demultiplexer. It enables to select the queue from the frames priority. Finally the goal of the last multiplexer representing one switch output port is to apply one scheduling policy between their different queues. The output capacity of the component 7 corresponds to the network bandwidth. The forwarding policy of the output multiplexers will be also chosen according the considered scheduling algorithm: Strict Priority or Weighted Fair Queuing. In the previous section, we have given bounded delays for crossing the output multiplexer for the two scheduling poli-

cies SP and WFQ (equation (4)). For the other components of the switch model, we will use delays given in [4].

## 6. Method

The maximum delay  $\overline{D}$  for crossing a switch depends on the leaky bucket parameters : the maximum amount of traffic  $\sigma$  that can arrive in a burst and the upper-bound  $\rho$  on the long-term average rate. Consequently, we need to know the  $(\sigma, \rho)$  envelop at each point of the network. As shown by the figure 4, the problem is that we only know the initial arrival curve  $(\sigma^0, \rho^0)$ . The other arrival curves, for example after crossing one switch  $(\sigma^1, \rho^1)$  have to be determined.



**Figure 4. Burstiness along the network.**

In order to resolve the evolution of the burstiness constraint of a flow, Cruz extends the previous method. For a system for which the arrival of data is constrained by  $b_{in}$  ( $R_{in} \sim b_{in}$ ) and for which the delay  $\overline{D}$  for crossing the system is finite ( $\overline{D} < +\infty$ ), he shows that the output of data is constrained by  $b_{out}$  ( $R_{out} \sim b_{out}$ ) as:

$$b_{out}(x) = b_{in}(x + \overline{D}) \quad (5)$$

which gives by using (1):

$$\sigma_{out} = \sigma_{in} + \rho_{in} \overline{D}, \quad \rho_{out} = \rho_{in} \quad (6)$$

In the case of the figure 4, the arrival curve after crossing the first switch will be also defined by  $(\sigma^1, \rho^1) = (\sigma^0 + \rho^0 \overline{D}_{switch}, \rho^0)$ .

This analysis is based on the fact that the arrival rate stays constant and that the delay is translated in a supplementary burst (*seconds in bits*). Since the routing strategy is fixed at any point of switched Ethernet networks, all of the previous upper-bounded delays are translated in upper-bounded output burstiness and the difference of burstiness

between the input and the output of the network will enable to determine end-to-end delays. All of these points are put together in the following algorithm. Its philosophy is to derive in a first time the delay equation in output burstiness equation, then to compute the output burstiness of each stream at each point of the network and finally to obtain an upper-bounded delay from the end-to-end burstiness difference. The method steps of the resolution are:

1. Identify all streams on each station and determine the initial leaky bucket values.
2. Identify the route of each stream.  
In switched Ethernet networks, paths are determined by the spanning tree protocol.
3. On each switch, formulate all streams output burstiness equations as described in the equation (6). By convenience, it is suggested to choose the notation  $\sigma_i^j$ , where  $i$  is the stream identifier and  $j$  is the number of crossed switches.
4. Define the equation system under the matrix equation:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} * \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

5. Calculate the burstiness values  $\sigma_i$ .
6. From equation (6), determine the bound

$$\overline{D}_i = \frac{\sigma_i^h - \sigma_i^0}{\rho_i}$$

$h$  represents the number of crossed switches.

## 7. Conclusion

The major result of this paper is a method to take into account the Classification of Service in the computation of upper-bounded end-to-end delays. The next step of this work will be to define an optimizing algorithm of quality of service offered by the network. This algorithm will ensure in priority the respect of time constraints of vehicle control applications and the bandwidth not used by the real-time applications will be given for entertainment software. Then, the problem will be to adjust this bandwidth in a better way in considering different class of service according to the type of entertainment application, to the importance of the passenger demands in order to control network burst, and almost to the pricing policy defined by the companies to access to internet, to upload video, to listen radio ...

## Acknowledgements

This work has been performed in the framework of the FP6 IST STREP project entitled "Networked Control Systems Tolerant to fault : NeCST" ([www.strep-necst.org](http://www.strep-necst.org)) which is funded by the European Community.

## References

- [1] M. Alves, E. Tovar, and F. Vasques. Ethernet goes realtime : a survey on research and technological developments. Technical report, HURRAY-TR-2k01, 2001.
- [2] R. Cruz. A calculus for network delay, part I : Networks elements in isolation. *IEEE Trans. on Information Theory*, 37:114–131, jan 1991.
- [3] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. In *Proceedings of ACM SIGCOMM'89*, pages 1–12, sep 1989.
- [4] J.-P. Georges, T. Divoux, and E. Rondeau. A formal method to guarantee a deterministic behaviour of switched Ethernet networks for time-critical applications. In *IEEE intl symp. on computer aided control system design*, sep 2004.
- [5] J.-P. Georges, T. Divoux, and E. Rondeau. Strict priority versus Weighted Fair Queueing in switched Ethernet networks for time-critical applications. In *Workshop on Parallel and Distributed Real-Time Systems*, Denver, Colorado, Apr. 2005.
- [6] IEEE. ANSI/IEEE Std 802.1D Media access control (MAC) bridges. IEEE Standards for local and metropolitan area networks, dec 1998.
- [7] IEEE. ANSI/IEEE Std 802.1Q Virtual bridged local area networks. IEEE Standards for local and metropolitan area networks, may 2003.
- [8] H. Kopetz, A. Damm, C. Koza, M. Mulazzani, W. Schwabl, C. Senft, and R. Zailinger. Distributed fault-tolerant real-time systems. *IEEE Micro*, 9(1):25–40, feb 1989.
- [9] J. Le Boudec and P. Thiran. *Network Calculus. A theory of deterministic queueing systems for the Internet*. Lecture notes in Computer Science. Springer Verlag, 2001. ISBN 3-540-42184-X.
- [10] G. Le Lann and N. Rivierre. Real-time communications over broadcast networks : the CSMA-DCR and the DOD-CSMA-CD protocols. Technical Report RR-1863, INRIA, mar 1993. <http://www.inria.fr/publication/publi-pdf/RR/RR-1863.pdf>.
- [11] A. Parekh and R. Gallager. A generalized processor sharing approach to flow control in integrated services networks : The single node case. In *IEEE/ACM Transactions on Networking*, volume 1-3, pages 344–357, jun 1993.
- [12] P. Pedreiras and L. Almeida. Flexibility, timeliness and efficiency over Ethernet. In *2002 International Workshop on Real-Time LANs in the Internet Age*, jun 2002.