This regular paper was presented as part of the main technical program at IEEE ETFA'2011

Bandwidth Mapping Algorithms in Distributed Media Control Applications

Javier Silvestre-Blanes
ITI. Universitat Politècnica de València
Ferrandiz y Carbonell. Alcoy 03801, Spain
jsilves@disca.upv.es

Paulo Pedreiras
UA / IT University of Aveiro
3810-193 Aveiro, Portugal
pbrp@ua.pt

Ricardo Marau
DEEC / IT University of Porto
4200-465 Porto, Portugal
marau@fe.up.pt

Abstract

Industrial media control applications require on-line QoS management services to dynamically allocate the bandwidth among the entities that compose the distributed application. The bandwidth distribution mechanism assigns bandwidth to each entity based on the QoS requirements and the current state of the network. However, mapping that bandwidth into operational parameters that serve the scheduling model of the network is not straightforward. Due to systems constraints, this is not a bijective function, therefore multiple solutions may mathematically represent the same bandwidth value but lead to different application performance levels. This paper presents two mapping algorithms and analyzes their performance in this context. The obtained results show that the selection of this mapping algorithm is highly relevant for the application performance.

1 Introduction and related work

Distributed multimedia applications are widely used in diverse domains, like home applications [2], wireless multimedia sensor networks [6][17], industrial applications, surveillance [10], etc. These media control applications (MCA) [4] can have soft and/or hard real-time requirements, depending on the specific application type. Typically, each video source i defines a set of quality of service (QoS) requirements (e.g. frame-rate, compression level), demanding appropriate bandwidth (w_i) reservation. On the network side, w_i is a strict bandwidth reservation that must be enforced and cannot be surpassed, with the penalty for interfering with other real-time data sources. In previous works [9], the authors used the dynamic QoS management features of the Flexible Time-Triggered communication over Switched Ethernet Protocol (FTT-SE) [5] to provide on-line real-time multimedia QoS management. In this framework the QoS management is multidimensional, permitting the dynamic adaptation of the compression rate and the allocated network bandwidth, according to the instantaneous application requirements.

The bandwidth assigned to each video stream is eventually mapped into channel operational parameters, i.e. to a given period (T) and image size (C). However, the conversion between bandwidth and network operational parameters is not univocal. In fact, different (C,T) tuples, assigned according to different mapping strategies, may correspond to the same bandwidth while leading to different application level behavior. Determining the mapping strategy that better suits a given application requirement is still an open question. Multimedia applications are very dependent of the scenario they are fitting to, so there is no general rule to specify how this mapping should be conducted.

The basic decision that must be handled by these mapping policies is whether it is better to use shorter periods with lower image sizes, or use larger periods and correspondingly larger image sizes. This trade-off has already been analyzed in the literature, for general multimedia scenarios using protocols such as MPEG to code video over a GOP (Group of Pictures, typically between 9 and 12 frames). In this general scenario, [13] and [12] the frame drops are selected according to the minimum distortion criterium, which requires storage of all the GOP before its transmission. In [1] control theory is used to avoid the over and under utilization of resources. As in the previous case, this method introduces and additional latency of one GOP. In [11] this trade-off is addressed analyzing only video compression, without considering the implied network challenges and their limitations.

Many Media Control Applications are delay sensitive, so the above methods are not adequate, since they cause additional delays. This paper presents and evaluates two different algorithms to map bandwidth onto network operational parameters in the scope of the dynamic QoS management framework previously developed by the authors

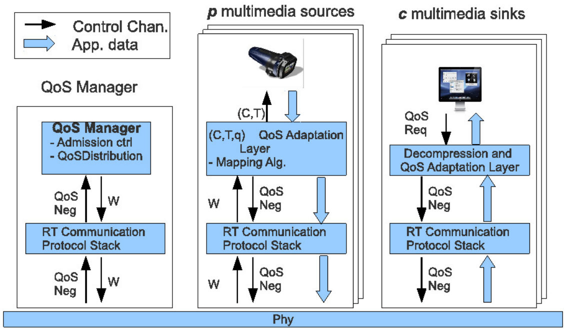

Figure 1. Generic System Architecture

[9]. Each one of the algorithms favors one of the operational components (minimize the period or maximize the image size). The rationale behind these algorithms is that for more dynamic image sources (with frequent changes in the scenario), it may be preferable to have a higher frame-rate, implying higher compression, while for more static image sources it can be better to use a lower frame rate and a correspondingly lower compression ratio.

The experimental evaluation method of the algorithms proposed in this paper follows the point of view of the Human Visual System (HVS). To obtain a quality evaluation of a set of videos there are subjective and objective methods [14]. In the subjective methods, the videos are shown to a set of observers who have to evaluate the quality, and after a statistical process, a quality measurement is obtained. This is the method that gives the best results according to the HVS but it is slow and expensive. The objective methods are based on the quality evaluation through equations based on parameters obtained from the original and distorted video (full reference). The most used metric is the Peak Signal to Noise Ratio (PSNR). As this has a poor correlation with the HVS [15], other metrics based on the measurement of structural error have been developed, such as SSIM (Structural Similarity Index Measurement) [16][8], which is also used as a metric in this paper.

The remainder of the paper is organized as follows. Section 2 presents the system and QoS models. Section 3 introduces the two mapping algorithms proposed in this paper. Section 4 presents the experimental results obtained with different stream types and network loads. Conclusions and future work are covered in Section 5.

2 System and QoS models

The generic system architecture considered in this paper is depicted in Fig.1. The central QoS manager is responsible for receiving the QoS requests for the diverse data sources, whether they are multimedia-related or not, and, in function of the instantaneous requests and available resources, allocates bandwidth to each real-time data source. Each data source comprises a QoS Adaptation Layer, which inter-operates with the QoS manager and fits the video stream data to the assigned bandwidth. This adaptation is done firstly by choosing appropriate quantification levels. The data source QoS Adaptation Layer may initiate a QoS renegotiation whenever the compression level falls out of the pre-defined range. Data sinks also comprise a QoS adaptation Layer and may also initiate QoS re-negotiations, e.g. in answer to application requests. Access to the communication channel is mediated by a real-time communication protocol, which is responsible for the timeliness of the communications.

2.1 System Model

The system model comprises a scenario with p distributed multimedia sources $\mathbb{M} \equiv \{M_i, i = 1..p\}$ which send streams to c sinks, through a real-time local area network. This network supports the requirements of dynamic distributed multimedia applications in industrial environments, namely dynamic traffic scheduling, online admission control and dynamic QoS management. Each multimedia source, at the network level, can be characterized by:

$$M_i^{net} = \{Pr_i, C_i, \mathbb{T}_i^j\}^{net} \tag{1}$$

where Pr_i is a priority that reflects the relative stream importance, $C_i \equiv \left[C_i^l, C_i^u\right]$ is the range of possible transmission buffer sizes (i.e. image frame size) and $\mathbb{T}_i^j, j = 1..n_i$ is the set of n_i transmission periods (i.e. the frame-rates) supported by the data source. The QoS system may change the quantification level q_i within $Q_i \equiv \left[q_i^l, q_i^u\right]$ to fit the images into the allocated buffer size. Values that fall outside of the compression range are undesirable since they lead to lower than desired image quality (compression too high) or to a inefficient utilization of the communication channel (compression too low). This adaptation is made based on the *q model* and the adaptation process shown in [9]. Through this model, the quantification level of current frame k (q^k) is used to estimate the cuantification value for the next frame (q^{k+1}). If the bandwidth falls inside a target window, then $q^{k+1} = q^k$, otherwise the q adaptation process is invoked.

The network communication protocol provides a total bandwidth U^S, which is shared among the p real-time data sources. Note that U^S is an upper-bound bandwidth that assures the timeliness of the real-time communication channels, depending e.g. on the particular scheduling policy employed. Furthermore, U^S is shared among all real-time data sources, whether they are multimedia-related or not. For the sake of simplicity, and without loss of generality, from this point on it is assumed that all data sources and sinks in the system are multimedia-related.

2.2 Bandwidth distribution

Each real-time data source demands a given desired bandwidth w_i^d, which depends both on the application demands and on the characteristics of the video streams. w_i^d

is constrained by M_i, i.e.:

$$\begin{aligned} w_i^{min} &= \frac{C_i^l}{max_{j=1..n_i}(T_i^j)} \\ w_i^{max} &= \frac{C_i^u}{min_{j=1..n_i}(T_i^j)} \\ w_i^{min} &\leq w_i^d \leq w_i^{max}, \forall i = 1..p \end{aligned} \quad (2)$$

The QoS manager firstly attempts to assign the desired bandwidth to all data sources. However, the channel capacity may not be sufficient to satisfy all the channel requests, i.e.:

$$\sum_{i=1..p} w_i^d = U_d > U^S \quad (3)$$

Whenever Equation 3 holds, the system is overloaded and it necessary to distribute the available bandwidth among the different data sources according to a given criterium. This procedure is carried out by a bandwidth distribution algorithm, which takes as inputs U^S, $\mathbb{M}$ and $\mathbb{W}^d = \{w_1^d..w_p^d\}$ and distributes U^S for each multimedia source. The output of the distribution algorithm is the set $\mathbb{W}^{qos} = \{w_1^{qos}, ..., w_i^{qos}, ..., w_p^{qos}\}$, where w_i^{qos} represents the bandwidth actually assigned to stream i.

Algorithm 2.1: $\mathbb{W}^{qos}$ = *wDistribution*$(\mathbb{M}, \mathbb{W}^d, U^S)$

```
comment: distributes the system bandwidth capacity
U_spare ← U^S − Σ_∀i w_i^min
Σ_∀i Pr_i = 1
for each M_i ∈ M, sorted by Pr_i
  do
    if (w_i^d − w_i^min) Pr_i < U_spare
      then w'_i ← (w_i^d − w_i^min) Pr_i
      else w'_i ← U_spare
    U_spare ← U_spare − w'_i
    w_i^qos ← w_i^min + w'_i
while (U_spare > ε)
return (W^qos = {w_1^qos, ..., w_n^qos})
```

Algorithm 2.1 shows a fixed priority-based policy implemented in this work, which is slightly different from the algorithm used in [9]. The algorithm starts by computing the amount of shareable bandwidth U_{spare}, i.e. the bandwidth remaining after satisfying the minimum requirements for all data sources. U_{spare} is then divided and assigned to the data streams according to their individual priorities. Starting from the one with higher priority, the excess bandwidth is computed (i.e., $w_i^d - w_i^{min}$) and weighted with the data stream priority. The demanded bandwidth is assigned to the data stream if not exceeding the system remaining capacity (U_{spare}). Otherwise, it is assigned the system remaining bandwidth. On each iteration the allocated bandwidth is subtracted to U_{spare}. The process repeates until the remaining bandwidth reaches a negligible value, i.e. is lower than (ε). Depending on the relation between the available and demanded bandwidth, some channels may get the requested bandwidth, others may get just their minimum bandwidth while others may get an intermediate value of bandwidth between the previous two cases.

2.3 Prototype platform

The mapping algorithms presented in this paper have been integrated in the Dynamic QoS management over FTT-SE framework [9], previously developed by the authors. This framework is based on the Flexible Time-Triggered (FTT) over Switched Ethernet (FTT-SE) protocol [3][5]. This is a research real-time protocol operating over Switched Ethernet networks, that supports synchronous (i.e. periodic) and asynchronous (i.e. sporadic) real-time traffic, as well as non-real-time traffic. The synchronous and asynchronous traffic is transmitted within separate windows with the former typically having priority over the latter. The non real-time traffic is scheduled in the background, within the asynchronous window. For the synchronous traffic, which is the only one relevant to the scope of this paper, a master/multi-slave transmission control technique is used, according to which a master addresses several slaves with a single poll message, considerably alleviating the protocol overhead when compared to the conventional master-slave techniques. The communication is organized in fixed duration slots called Elementary Cycles (ECs). Each EC starts with one poll message sent by the master, called Trigger Message (TM). The TM contains the schedule for that particular EC. Only the messages that fit within an EC are scheduled by the master, thus memory overflows inside the switch are completely avoided for such kinds of traffic.

The traffic is scheduled dynamically, i.e. for each EC the Master checks the properties of the messages active at that particular instant and schedules them. Therefore, changing the message set (adding, removing or changing the properties of messages) only requires updating the message properties database. To prevent changes that jeopardize the timeliness of the real-time traffic, all change request are subject to an admission control and rejected, if necessary. Finally, the Master also integrates a QoS manager. QoS changes are firstly checked for feasibility. If feasible, appropriate QoS distribution algorithms are applied and the message set updated. Thanks to the centralized architecture, the operations above mentioned are local to the Master node, which simplifies the process considerably. This protocol matches the requirements of dynamic distributed multimedia applications in industrial environments stated in Section 2.1. Further details about the protocol and, in particular, the QoS management subsystem, can be found in [9].

3 Bandwidth to (C,T) mapping

Algorithm 3.1: $\{(C_i, T_i)\forall_i\}$ = *uCTmap*($\mathbb{W}^{qos}, \mathbb{M}$)

comment: $succ(T_i)$ is the successor of T_i in the monotonically increasing set $\mathbb{T}_i^{FTT}$

for each $M_i \in \mathbb{M}$

do $\begin{cases} T_i = \max\left\{T_i^j, \forall_{j=1..n_i} : T_i^j \leq C_i^u / w_i^{qos}\right\} \\ C_i = w_i^{qos} * T_i \\ \textbf{if } C_i < C_i^l \\ \quad \textbf{then } \begin{cases} T_i = succ(T_i) \\ C_i = C_i^u \end{cases} \\ w_i = \frac{C_i}{T_i} \end{cases}$

return $((C_i, T_i)\forall_i, \mathbb{W} = \{w_1, \ldots, w_n\})$

The bandwidth allocation Algorithm 2.1 assigns to each data source i a given bandwidth w_i^{qos}. However, this bandwidth must be converted into network operational parameters, that is, into a tuple (C_i, T_i^j), which corresponds to a given image frame-rate and image size. This tuple is is computed in the QoS Adaptation Layer to conform the data generated with the channel characteristics.

In the general case this conversion is not univocal, possibly existing different (C_i, T_i) pairs that satisfy $w_i^{qos} \geq C_i/T_i$. In practical terms, different frame-rates and image sizes may generate the same bandwidth. While mathematically equivalent, these tuples may impact on the application performance, and thus it is necessary to develop appropriate mapping algorithms. Each of the algorithms introduced in this section favors one of the parameters, C or T, in order to enable the study of its influence in the transmitted video quality. Intuitively, adopting lower period values, i.e. using higher frame-rate, can be advantageous in the case of more dynamic images, since lower frame-rates may lead to the loss of important features of the video stream. Conversely, for more static images, i.e. when the images do not change significantly over time, it may be more useful to the application to use a lower frame-rate and transmit each individual image with higher quality.

Algorithm 3.2: $\{(C_i, T_i)\forall_i\}$ = *uCTmap*($\mathbb{W}^{qos}, \mathbb{M}$)

for each $M_i \in \mathbb{M}$

do $\begin{cases} T_i = \min\left\{T_i^j, \forall_{j=1..n_i} : T_i^j \geq C_i^l / w_i^{qos}\right\} \\ C_i = w_i^{qos} * T_i \\ \textbf{if } C_i > C_i^u \\ \quad \textbf{then } C_i = C_i^u \\ w_i = \frac{C_i}{T_i} \end{cases}$

return $((C_i, T_i)\forall_i, \mathbb{W} = \{w_1, \ldots, w_n\})$

Mapping Algorithm 3.1, which was originally used in [9], attempts to maximize C_i. To achieve this goal, C_i is fixed at the highest possible value and then the highest T that satisfies or exceeds the assigned bandwidth w_i^{qos} is computed. However, since $\mathbb{T}$ is a discrete set, there might not exist an exact match solution. In this case T_i is lower than desired and thus C_i is recomputed accordingly, since the assigned bandwidth cannot be exceeded. If the computed C_i violates the lower bound (C_i^l), then a higher period must be selected and the highest frame size is used. Note that in this latter case $w_i < w_i^{qos}$. This procedure is executed for all data sources. The interested reader can resort to [9] for a more detailed explanation of this algorithm. Mapping Algorithm 3.2 attempts to maximize the frame-rate, i.e., use the minimum T_i. To achieve this goal, firstly it computes the period T_i that fits best, without exceeding, w_i^{qos}, considering the minimum possible frame-size (C_i^l). As in the previous algorithm, the discrete nature of periods may not allow to obtain a period that exactly matches w_i^{qos} for C_i^l. In that case C_i is recomputed, to allow a full use of the assigned bandwidth. Also in this case it may happen that, as a result of the previous computation, the constraints on C_i are violated. In that case C_i is assigned with the highest possible C_i^u, to approach the allocated bandwidth as much as possible, but $w_i < w_i^{qos}$.

4 Experimental results

This section presents simulation results that evaluate the relative performance of the mapping algorithms presented in Section 3 with different types of video streams. The simulation results are based in the simulation environment described in Section 2.3, which emulates the behavior of the FTT-SE protocol.

4.1 Video stream characterization

To analyze the differences in the bandwidth mapping algorithms, six different streams with different QoS parameters are used [7][9]. These streams are representative of industrial monitoring applications, showing robots and other industrial tasks that represent different video applications requirements. These are medium resolution streams, captured at 25 frames per second (T=40 ms), having each one 8,000 frames. Details of how streams are captured, and particular aspects of the scenes can be found in [7]. The stream QoS parameters used in the experiments are shown in Table 1. The full specification of the QoS parameters includes ranges for q, T and C, as well as the (fixed) priority Pr. The streams period varies between T^l and T^u, with a resolution of $40ms$.

Video streams can be classified as high motion or low motion, depending on how often significant changes occur in the scenes. As the trade-off between C and T in the bandwidth mapping can be influenced by the stream motion class, it is necessary to classify the streams. Streams M_1, M_3 and M_4 are considered to be high motion streams. These streams are captured in a car production plant, showing assembly robots. The movement of the

Table 1. Stream properties for experiments

	M_1	M_2	M_3	M_4	M_5	M_6
q^l	30	25	20	25	20	20
q^u	70	85	85	75	70	60
T^l(ms)	40	40	40	80	40	80
T^u(ms)	400	400	280	240	400	240
C^l(kB)	25	20	30	20	30	15
C^u(kB)	90	70	80	90	95	60
Pr	0.21	0.20	0.19	0.18	0.12	0.10
Motion	H	L	H	H	L	L

robots and their operations, like welding pieces, present significant changes from frame to frame. Streams M_2, M_5 and M_6 are classified as low motion streams, since they are captured by fixed cameras and the industrial scenes monitored have generally low activity.

In the experiments U^S varies from 5Mbps to 25 Mbps, in 5 Mbps steps, to model different load conditions. In each experiment a total of 240,000 frames was transmitted. Image size, period and compression level, as well as SSIM, PSNR and Dropped Frames (DrF) are computed both individually and in average. SSIM is considered by recent studies as the metric with the strongest correlation with HVS. It is necessary to mention here, that a difference of 0.01 in this index can represent around 1dB in PSNR measurement, so small differences in SSIM figures could represent important differences from the point of view of the video quality.

4.2 Results Analysis

Table 2 contains the results obtained with Algorithm 3.1 (Exp. 1), while Table 3 depicts the results obtained with Algorithm 3.2 (Exp. 2). The first general comment that can be made is that, as expected, $\hat{c}$ (mean size C of frames for one stream, measured in kilo-bytes) is higher in Exp. 1 than in Exp. 2, whereas $\hat{t}$ (mean T for frames for one stream, measured in miliseconds) is lower in Exp. 2 than in Exp. 1. Therefore, each one of the algorithms in fact favors the parameter it is intended to. It can also be observed that the mean quantification factor $\hat{q}$ is lower in Exp. 2 than in Exp. 1. This behavior was also expected since, for the same bandwidth, using a lower T implies reducing C, which is achieved with a reduction in q. Therefore, both algorithms produced results consistent and logically compatible with the expected behavior.

The differences between $\hat{c}$ and $\hat{t}$ are quite significant in all scenarios except for the U^S= 5Mbps case. When U^S=5Mbps, the system is heavily overloaded, therefore the solution space is very limited and consequently both algorithms produce very similar results. Only stream M_2 has a more noticeable difference. As this stream is classified as low motion, the increase in the transmission period does not compensate the reduction in the size that the system can use for their codification, so Algorithm 3.1 give slightly better results.

Another boundary scenario is for U^S=25Mbps. In this case the available bandwidth is almost enough to satisfy all the stream demands. The differences are not dramatic but in some cases are still noticeable. Low motion streams, like M_2 and M_6, have better results with Algorithm 3.1, increasing the quality in 2 and 1 dB for PSNR, and around 0.02 in SSIM metric. High speed streams M_1 and M_4 perform better with Algorithm 3.2, with a difference around 0.5dB and 1dB in the PSNR, and around 0.01 in the SSIM metric. M_3 performed contrarily to the expectations, since it is a high motion stream but performed better with Algorithm 3.1. It can be seen that both algorithms assigned to this stream the very same average period (80ms), but Algorithm 3.1 assigned an higher image size. Nevertheless the performance degradation is residual.

For U^S=10Mbps and U^S=15Mbps there are noticeable differences in the stream set, except for M_1. Being this one highest priority stream, both algorithms give similar (C,T) values. For U^S=20Mbps the algorithms perform differently for all messages. Regarding the quality metrics, we saw a clear tendency for each algorithm to perform better in one or the other scenario, according with the type of the streams. There are however a few exceptions, which can be justified with the non linearity of the mapping (periods are discrete and the image size is constrained). To have a quantitative idea, lets us to define a threshold of relevance of 0.005 for SSIM and 0.25dB for PSNR, and consider together $U^S = 10Mbps, 15Mbps, 20Mbps$. For the PSNR metric, in 10 out of the 18 cases, the algorithm perform better in the stream they are supposed to, in 5 cases there is an inversion and in 3 cases the difference is not relevant. Regarding SSIM, in 9 cases the algorithms favor the scenarios they are supposed to, in 2 cases there is an inversion and in 7 cases the difference is not relevant.

4.3 Hybrid algorithm

The QoS management model enables the combination of different mapping algorithms. To explore this capacity, we realized Experiment 3, in which Algorithm 3.1 is used for streams M_2, M_5 and M_6 whereas Algorithm 3.2 is used for the other streams. The results obtained are shown in Table 4. We saw a general tendency for the adaptive algorithm to approach the results that can be achieved with the more suitable algorithm, i.e., the adaptive algorithm gives the best results. Globally these experiments show similar results for U^S=10Mb and a slight improvement for U^S=15Mb and U^S=25Mb. However, for U^S=20Mb this slight improvement is for Alg. 3.1.

These results are influenced by the number of frame drops produced, which has an important influence on the video quality. It is expected that lower values of T increase the probability of frame drops, and this rule is satisfied generally in Exp. 1 and Exp. 2. However, even relatively similar values of T can produce different global results as changes on one stream may impact on all the other streams, as in stream M_1 for U^S=25 with Alg. 3.1. This

Table 2. Results for exp. 1 (Alg.3.1)

U^S=5	$\hat{c}$(kB)	$\hat{t}$(ms)	$\hat{q}$	SSIM	PSNR	DrF
M_1	36.4	400	55	0.873	27.26	15
M_2	28.4	400	46	0.909	29.87	9
M_3	36.0	280	37	0.873	26.78	28
M_4	25.6	240	33	0.850	26.70	35
M_5	29.9	400	56	0.912	30.89	3
M_6	20.8	240	23	0.894	28.50	22
mean	29.5	326	41	0.880	28.33	18
U^S=10	$\hat{c}$(kB)	$\hat{t}$(ms)	$\hat{q}$	SSIM	PSNR	DrF
M_1	36.3	128	56	0.9121	30.14	27
M_2	46.5	204	75	0.929	31.98	2
M_3	53.4	240	79	0.902	29.41	7
M_4	45.5	240	69	0.868	27.65	13
M_5	36.9	394	68	0.915	31.14	1
M_6	24.2	240	29	0.906	29.06	24
mean	40.5	241	62.7	0.905	29.89	12.3
U^S=15	$\hat{c}$(kB)	$\hat{t}$(ms)	$\hat{q}$	SSIM	PSNR	DrF
M_1	37.6	80	56	0.930	31.79	33
M_2	44.2	120	73	0.938	31.72	2
M_3	53.4	160	78	0.920	31.76	27
M_4	45.1	200	69	0.876	28.28	24
M_5	31.9	235	60	0.920	31.80	3
M_6	32.5	240	48	0.920	29.26	1
mean	40.7	172.6	64	0.917	31.11	15
U^S=20	$\hat{c}$(kB)	$\hat{t}$(ms)	$\hat{q}$	SSIM	PSNR	DrF
M_1	41.8	80	65	0.935	32.24	32
M_2	40.0	80	72	0.946	34.00	2
M_3	40.5	80	59	0.952	35.09	27
M_4	45.2	160	70	0.887	29.04	22
M_5	34.4	154	63	0.928	32.68	4
M_6	36.4	240	58	0.923	30.25	0
mean	39.7	132	64	0.929	32.21	14.5
U^S=25	$\hat{c}$(kB)	$\hat{t}$(ms)	$\hat{q}$	SSIM	PSNR	DrF
M_1	31.9	42	45	0.944	33.17	541
M_2	50.5	80	78	0.950	34.47	1
M_3	54.2	80	80	0.954	37.34	22
M_4	36.4	123	56	0.895	29.59	41
M_5	32.1	120	60	0.930	32.93	2
M_6	35.8	240	56	0.923	30.23	0
mean	40.15	114.3	62.5	0.933	32.95	101

Table 3. Results for exp. 2 (Alg.3.2)

U^S=5	$\hat{c}$(kB)	$\hat{t}$(ms)	$\hat{q}$	SSIM	PSNR	DrF
M_1	34.8	347	20	0.874	27.27	42
M_2	22.1	343	30	0.898	29.53	13
M_3	36.0	279	37	0.873	26.81	42
M_4	25.9	239	33	0.851	26.72	41
M_5	28.0	399	51	0.911	30.82	3
M_6	21.0	240	23	0.895	28.54	30
mean	27.9	307	28	0.884	28.28	28
U^S=10	$\hat{c}$(kB)	$\hat{t}$(ms)	$\hat{q}$	SSIM	PSNR	DrF
M_1	37.8	134	57	0.908	29.73	85
M_2	23.2	120	33	0.917	31.21	13
M_3	37.1	160	42	0.905	29.89	46
M_4	29.6	159	42	0.874	28.17	48
M_5	27.2	240	49	0.918	31.8	3
M_6	22.8	200	26	0.922	30.47	30
mean	29.6	168.8	41.5	0.907	30.21	37.5
U^S=15	$\hat{c}$(kB)	$\hat{t}$(ms)	$\hat{q}$	SSIM	PSNR	DrF
M_1	36.1	80	53	0.929	31.56	29
M_2	26.9	80	42	0.931	32.42	8
M_3	38.8	120	51	0.931	32.79	13
M_4	31.3	120	46	0.892	29.34	37
M_5	30.8	194	58	0.922	32.03	4
M_6	22.4	160	26	0.906	29.49	15
mean	31.05	125.6	46	0.919	31.2	17,6
U^S=20	$\hat{c}$(kB)	$\hat{t}$(ms)	$\hat{q}$	SSIM	PSNR	DrF
M_1	32.1	53	45	0.936	32.55	152
M_2	29.5	64	48	0.937	33.12	3
M_3	46.2	106	61	0.926	32.6	115
M_4	32.3	93	47	0.901	30.04	101
M_5	30.1	153	56	0.925	32.37	4
M_6	21.1	120	23	0.904	29.52	137
mean	31.8	98.16	46.7	0.922	31.7	85
U^S=25	$\hat{c}$(kB)	$\hat{t}$(ms)	$\hat{q}$	SSIM	PSNR	DrF
M_1	34.07	40	49	0.952	33.6	188
M_2	23.05	40	33	0.933	32.76	18
M_3	43.4	80	66	0.950	34.47	46
M_4	30.7	80	44	0.909	30.6	61
M_5	28.9	113	53	0.928	32.8	1
M_6	21.2	120	23	0.905	29.67	35
mean	30.2	78.8	44.7	0.930	32.31	58

also explains the differences between Exp. 1 and Exp. 3.

4.4 Final remarks on the experimental results

The intensive experimentation done confirms that the performance of mapping algorithms is indeed correlated with the stream properties. However, the observed differences are not dramatic. This is mainly due to the contents of the high motion streams used in the experiments. These streams were obtained in real plants and only exhibit high motion contents during certain time periods. Figure 2 shows the evolution of C (bytes), T (msec.) and q for each frame of high-motion stream M_3. For comparison purposes, Figure 3 shows the C, T and q for each frame of low-motion stream M_5. As can be seen, strong changes in C of stream M_3 only occur during certain periods of time. For extended time periods, the behavior of stream M_3 resembles M_5, i.e., is essentially low-motion. In this sense, the use of Algorithm 3.2 during the whole stream duration is self-defeating.

Summing up, we can conclude that mapping algorithms in fact may have a relevant impact on multimedia transmission over real-time networks. However, the extent of such impact is strongly dependent on the nature of the media being transmitted. Furthermore, in real scenarios the classification can be hard, since streams may exhibit different behavior over time. One possible way of tackling this issue is classifying online the streams and selecting dynamically the most adequate algorithm. This will be addressed in future work.

Table 4. Results for exp. 3

U^S=5	$\hat{c}$(kB)	$\hat{t}$(ms)	$\hat{q}$	SSIM	PSNR	DrF
M_1	35.1	340	51	0.875	27.29	40
M_2	23.7	400	34	0.902	29.43	9
M_3	36.5	279	39	0.876	27.01	35
M_4	26.2	239	34	0.852	26.77	39
M_5	28.3	400	52	0.911	30.83	3
M_6	20.7	240	22	0.893	28.48	25
mean	28.4	316	39	0.885	28.3	25
U^S=10	$\hat{c}$(kB)	$\hat{t}$(ms)	$\hat{q}$	SSIM	PSNR	DrF
M_1	35.3	120	52	0.911	30.03	80
M_2	49.5	240	77	0.927	31.72	2
M_3	37.4	160	44	0.909	30.14	33
M_4	31.3	156	46	0.878	28.37	42
M_5	36.6	400	68	0.915	31.11	1
M_6	22.9	240	27	0.901	28.85	27
mean	35.5	219	52	0.907	30.09	30
U^S=15	$\hat{c}$(kB)	$\hat{t}$(ms)	$\hat{q}$	SSIM	PSNR	DrF
M_1	37.5	80	56	0.931	31.87	33
M_2	41.5	120	70	0.937	32.9	2
M_3	39.6	120	55	0.933	32.98	14
M_4	32.4	120	48	0.893	29.42	42
M_5	35.4	275	66	0.920	31.69	0
M_6	30.8	240	45	0.918	29.83	20
mean	36.2	159.1	56	0.922	31.48	18
U^S=20	$\hat{c}$(kB)	$\hat{t}$(ms)	$\hat{q}$	SSIM	PSNR	DrF
M_1	32.0	49	45	0.939	32.78	375
M_2	34.6	81	59	0.940	33.29	2
M_3	37.9	81	45	0.932	33.12	322
M_4	29.2	80	41	0.893	29.74	310
M_5	33.2	195	62	0.923	32.15	3
M_6	35.8	240	57	0.934	30.22	2
mean	33.78	121	51	0.927	31.88	169
U^S=25	$\hat{c}$(kB)	$\hat{t}$(ms)	$\hat{q}$	SSIM	PSNR	DrF
M_1	35.5	40	52	0.954	33.86	181
M_2	45.5	80	74	0.947	34.17	2
M_3	45.5	80	67	0.949	34.58	34
M_4	30.6	80	44	0.909	30.61	54
M_5	34.3	154	63	0.927	32.56	2
M_6	36.3	240	58	0.923	30.25	0
mean	37.9	112	59	0.935	32.67	45

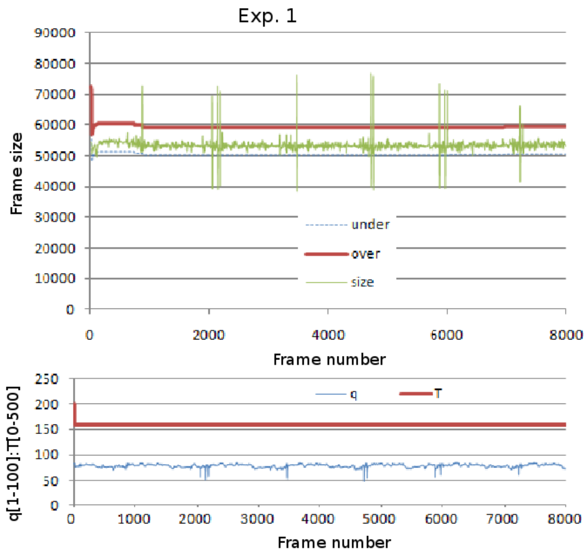

Figure 2. M_3 **results for Exp. 1**

5 Conclusions

The transmission of multimedia streams with real-time requirements demands appropriate network capabilities. [9] presents a multidimensional dynamic QoS adaptation mechanism which changes dynamically the channel bandwidth and quantification factor according to streams needs and overall system load.

The bandwidth allocated by the bandwidth distribution algorithm has to be mapped into the network operational parameters (C,T). This mapping is not univocal and different strategies have an impact on the application-level performance. This paper presents two different mapping algorithms. One favors the use of higher image sizes and should perform better in static scenarios, while the other algorithm favors higher frame-rates and should be more beneficial in high-motion scenarios.

Both algorithms have been intensively tested using 6 different industrial video streams, with different motion types. Both single mapping algorithm (i.e. all streams handled by one of the algorithms) and hybrid mapping algorithm (streams handled by the most adequate algorithm) have been analyzed. The results obtained show PSNR differences up to around 2dB, or 0.02 in SSIM obtained in different streams depending on the mapping algorithm used.

The classification of the streams proved to be difficult to deal with, since video streams are frequently heterogeneous, exhibiting activity peaks mixed with stability periods. This heterogeneity somehow dilutes the observable differences. This introduces a new dimension for the QoS management, which is the development of a live motion detection mechanism to select the appropriate mapping algorithm depending on the current scene classification. This will be addressed in future work. Moreover, we are also working in the compilation of a new industrial video database, which can represent better the special properties of videos in this domain.

Acknowledgments

This work is supported by the MCYT of Spain under the project TIN2010-21378-C02-02 and by FEDER through " Programa Operacional Factores de Competitividade COMPETE" jointly with the Portuguese "FCT Fundação para a Ciência e a Tecnologia" in the scope of project PTDC/EEA-ACR/73307/2006 and the iLAND project, call 2008-1 of the EU ARTEMIS JU Programme.

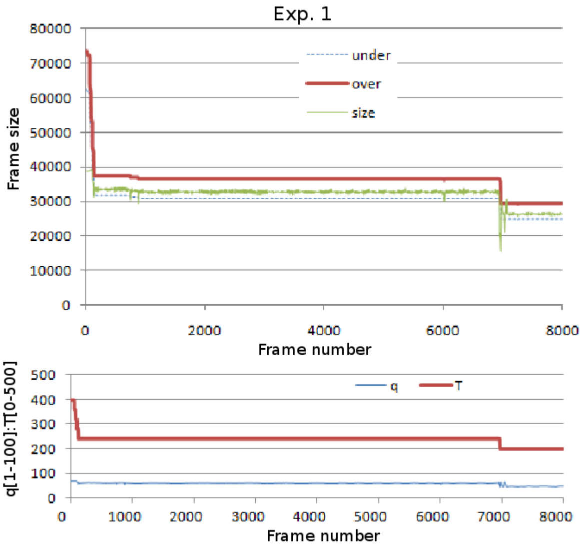

Figure 3. M_5 **results for Exp. 1**

References

[1] Rodrigo F. Coelho, Anand Kotra, and Gerhard Fohler. A control theory approach to video stream adaptation for restricted bandwidth networks. In *Proc. of First Int. Workshop on Adaptive Resource Management*. Artist, April 2010.

[2] G. Giorgetti et al. An integrated solution for home automation. In *Int. Symp. on Consumer Electronics*, pages 1–4, Vilamoura , portugal, April 2008. IEEE.

[3] Ricardo Marau, Luís Almeida, and Paulo Pedreiras. Enhancing real-time communication over COTS Ethernet switches. In *Proc. of 6th Int. Workshop on Factory Communication Systems (WFCS'06)*, pages 295–302, Torino, Italy, 2006. IEEE.

[4] F.G. Molinero. Real-time requirements of media control applications. In *19th Euromicro Conference on Real-Time Systems*, June 2007.

[5] P. Pedreiras, P. Gai, L. Almeida, and G. Buttazzo. FTT-Ethernet: A flexible real-time communication protocol that supports dynamic QoS management on ethernet-based systems. *IEEE Transactions on Industrial Informatics*, 1(3):162–172, 2005.

[6] Scott Pudlewski and Tommaso Melodia. A distortion-minimizing rate controller for wireless multimedia sensor networks. *Computer Communications*, 33(12):1380–1390, July 2010.

[7] J. Silvestre, V. Sempere, and T. Albero. Industrial video sequences for network performance evaluation. In *Proc. Int. Workshop on Fact. Comm. Sys.*, pages 343–347. IEEE, Sep. 2004.

[8] Javier Silvestre-Blanes. Structural similarity image quality reliability. determining parameters and window size. *Signal Processing*, 91:1012–1020, 2011.

[9] Javier Silvestre-Blanes, L. Almeida, R. Marau, and P. Pedreiras. On-line QoS management for multimedia real-time transmission in industrial networks. *IEEE Transactions on Industrial Electronics*, 58(3):1061–1071, Mar. 2011.

[10] G.S.V.S. sivaram, Mohan S. Kankanhalli, and K.R. Ramakrishnan. Design of multimedia surveillance systems. *ACM Transactions on Multimedia Computing, Communications and Applications*, 5(3), August 2009.

[11] Huey-Min Sun, Yung-Chuan Lin, and LihChyun Shu. The impact of varying frame rate and bit rate on perceived quality of low/high motion sequences with smooth/complex texture. *Multimedia Systems*, 14(1):1–13, 2008.

[12] Wei Tu, Wolfgang Kellerer, and Eckehard Steinbach. Rate-distortion optimized video frame dropping on active network nodes. In *Proc. Int. Packet Video Workshop*, 2004.

[13] Anthony Vetro, Yao Wang, and Huifang Sun. Rate-distortion optimized video coding considering frameskip. In *Int. Conf. on Image Processing*, pages 534–537, Thessaloniki , Greece, Oct. 2001. IEEE.

[14] Zhou Wang and A. C. Bovik. *Modern Image Quality Assessment (Synthesis Lectures on Image, Video, & Multimedia Processing)*. Morgan & Claypool Publishers, San Rafael, CA, 2006.

[15] Zhou Wang and A. C. Bovik. Mean squared error: Love it or leave it? *IEEE Signal Processing Magazine*, pages 98–11, 2009.

[16] Zhou Wang, Ligan Lu, and A. C. Bovik. Video quality assessment based on structural distortion measurement. *Signal Processing:image communication*, 19:121–135, 2004.

[17] Ruonan Zhang, Lin Cai, Jianping Pan, and Xuemin Shen. Resource management for video streaming in ad hoc networks. *Ad Hoc Networks*, 9:623–634, 2011.