Mixed Criticality Scheduling Applied to JPEG2000 Video Streaming Over Wireless Multimedia Sensor Networks

Alemayehu Addisu
Dip. di Informatica e Elettronica
Politecnico di Torino
C.so Duca degli Abruzzi, 24
10129 Torino, Italy
Email: alex.addisu@gmail.com

Laurent George
LIGM, Cite Descartes,
Copernicus Bat, 5 bd Descartes,
77454, Champs sur Marne,
France
Email: lgeorge@ieee.org

Vincent Sciandra
Transdev,
32 boulvard Gallieni
92130, Issy-les-Moulineaux,
France
Email: sciandra@ece.fr

Max Agueh
LACSC ECE,
37, Quai de Grenelle
75725 Paris Cedex 15,
France
Email: agueh@ece.fr

***Abstract*—In this paper, we propose a mixed-criticality scheduling scheme for selection of JPEG2000 codestream features to be transmitted over Wireless Multimedia Sensor Networks (WMSNs). We extend the application of mixed-criticality scheduling model to the wireless domain. We show that by adopting mixed-criticality scheduling scheme, an improved end-to-end response time is gained with respect to the classical case where all information exhibit the same level of importance.**

I. Introduction

Thanks to the integration of inexpensive complementary metal-oxide semiconductors (CMOS) cameras and microphones, WMSNs are emerged as interesting framework for applications such as enhanced surveillance and monitoring systems. These networks are capable of sensing multimedia content including audio, still images and videos in addition to scalar sensor data (e.g. temperature, humidity, etc ...) [1].

Several surveys were conducted on different aspects of wireless multimedia sensor networks. For example, [1] and [2] clearly highlights the state-of-the-art and main research challenges for development of WMSNs. These surveys discussed several characteristics of WMSNs, among which application scenarios, existing solutions and different open research issues. Since WMSNs are resource constrained networks (e.g limited computation power, reduced memory, narrow bandwidth, etc ...), image and video transmission over such networks is still an important challenge which needs to be addressed.

In this context, mixed-criticality scheduling is used to arbitrate flows generated from different sources. Each source uses JPEG2000 compression algorithm to encode images with multi-layer and multi-resolution features of JPEG2000. During transmission of image/video from the sources, all information do not possess the same level of criticality. Information from one source is more critical than that of other source depending on the available channel capacity. Hence, we can model our system based on the mixed-criticality properties.

The mixed-criticality nature of the system arises from the fact that while we would like to transmit all information (all layers and resolutions) under high availability of the bandwidth. However, it is important that the critical information has to be transmitted even when the bandwidth is low. We consider a communication channel with *L*-levels of bandwidth values and transmission of periodic images that possess different levels of criticality. Furthermore, the model is considered as a non-preemptive scheduling problem, in the sense that once the transmission of an image is started, it cannot be preempted by another with higher criticality.

Due to rapidly increasing cost, power and thermal dissipation constraints, there is an increasing trend in embedded system towards implementing multiple functionalities upon a single shared computing platform. Typically, all these different functionalities do not possess the same level of criticality to the overall system performance. This concept of mixed-criticalities gave rise to a mixed-criticality scheduling problem. It is initially introduced by Vestal [3]. He pointed out that there is a difficulty in computing the exact Worst Case Execution Times (WCETs). The more conservative the approximation of WCETs, the high level of assurance that the execution of the task never exceeds it WCET. This confidence levels form the basis for different levels of rigorousness (level of criticality) of the system. To solve the problem of mixed-criticality scheduling, he also suggested a fixed-task-priority strategy based on ”Audsley approach” [4]. Based on this approach, other authors provide an improved way to tackle the intractability of mixed-criticality scheduling. For instance, Baruah et. al. [5] proposed an algorithm called Own Criticality Based Priority (OCBP) to schedule a mixed-criticality system with a finite number of jobs. In their work, they showed that OCBP-schedulability offers performance guarantee that is superior to performance guarantee offered by the Worst Case Reservations (WCR) schedulability. In [6], the authors proposed another algorithm, called Priority List Reuse Scheduling (PLRS) to schedule certifiable mixed-criticality sporadic tasks system. They used fixed-job-priority scheduling scheme and assigned job priorities by exploring and balancing the asymmetric effects between the workload of different criticality levels.

Through simulation, they found that the run-time complexity of PLRS is polynomial time.

In our work, we model the transmission channel as a non-preemptive uniprocessor with limited amount of transmission time. The channel has *L* levels of speed, which corresponds to the amount of available bandwidth in the network. We can represent *B(l)* as the available bandwidth at criticality level *l*∈ [1,*L*]. Hence, the transmission time $C_i(l)$ can be seen as the time it takes to send data units at a rate of *B(l)* and criticality level *l*.

The remainder of this paper is organized as follows. In section 2, we discuss mixed-criticality scheduling model along with some overviews on JPEG2000 compression algorithm and available bandwidth estimation tools. Then in section 3, the application of mixed-criticality scheduling to video streaming system is given. In section 4, implementation of the proposed algorithm is detailed. Experiments and results are provided in section 5. Then, conclusion and future work are followed in section 6.

II. Model, Definitions and Overviews

A. Mixed-criticality Scheduling

1) ***MC tasks and jobs :*** We consider the scheduling of Mixed-criticality(MC) tasks on a non-preemptive single processor with task sets $\tau = \{\tau_1, \tau_2, ..., \tau_n\}$. Furthermore, the maximum criticality of a task is bounded by *L*. Each MC task is characterized by 5-tuple $\tau = \{R_i, T_i, D_i, \chi_i, C_i\}$ where:

- $R_i \in \mathbb{N}$ is the release time of the first job of task τ_i
- $T_i \in \mathbb{N} \setminus \{0\}$ is the period of task τ_i
- $D_i \in \mathbb{N} \setminus \{0\}$ is the deadline of task τ_i, $D_i \leq T_i$
- $\chi_i \in \mathbb{N}$ is the maximum criticality of the task τ_i, $\chi_i \leq L$
- $C_i \in \mathbb{N}^{\mathbb{L}}$ is a size *L* vector of WCETs, where $C_i(l)$ is an estimation of the WCET of task τ_i at criticality level $l \in$ [1,*L*]

We assume $C_i(l)$ is monotonically non-decreasing for increasing *l*. More precisely, for task τ_i:

- $\forall\ m \in [1,\ \chi_i] : C_i(m) \leq C_i(m+1)$
- $\forall\ m \in [\chi_i,\ L] : C_i(m) = C_i(\chi_i)$

A *job* in MC system is characterized by a 5-tuple of parameters: $J_j = \{r_j, d_j, \chi_j, C_j, c_j\}$ where:

- $r_j \in \mathbb{N}$ is the release time of the job J_j,
- $d_j \in \mathbb{N}$ is the absolute deadline,
- $\chi_j \in \mathbb{N}^+$ is the criticality of the job,
- $C_j \in \mathbb{N}^{\mathbb{L}}$ is a size *L* vector of WCETs of J_j,
- $c_j \in \mathbb{N} \setminus \{0\}$ is the exact execution of the job J_j.

The idea of the MC job model is: job J_j is released at time r_j, has deadline at d_j, and needs to execute for some amount at time c_j. However, the value of c_j is not known beforehand, but only becomes revealed by actually executing the job until it signals that it has completed execution.

At any time, we call a job is *available* if its release time has passed and the job has not signalled execution completion. Let us define a notion of a *scenario*. Each job J_j requires an amount of execution time c_j within $[r_j,\ d_j]$. We call a collection of execution times *S* = $\{c_1, c_2, ..., c_n\}$ a *scenario* and it consists of *n* jobs.

The criticality level of a scenario *S* can be defined as the smallest integer *l* such that $c_j \leq C_j(l)\ \forall j$. If no such *l* exists, then the scenario is said to be *erroneous*, since at least one task exceeds its WCET at its own criticality.

2) ***MC-schedulability:*** In literature, it is shown that MC-scheduling problem is NP-hard in strong sense. For example in [5], Baruah et al. proved that when MC is applied to a finite set of jobs, it is not possible to find a solution in polynomial time. This condition forces the research community to come up with a *sufficient* condition that can be verified in polynomial time. In [3], Vestal determined a total ordering of the tasks in τ offline. Each task is assigned a distinct priority and jobs inherit the priority of the task that released them. At each moment, the scheduler dispatches the available job with the highest priority. The priority assignment is realized using "Audsley approach" based on the following definition with assumption of priority *n* be the lowest priority and 1 be the highest priority.

Definition: A task τ_i in a mixed-criticality task set τ is said to be *viable at the lowest priority level* if all of the following conditions hold true:

1) the lowest priority is assigned to τ_i,
2) all other tasks in τ can be assigned any priority provided that these priorities are higher than the priority assigned to τ_i and
3) every job released by τ_i meets its deadline when it is executed for at most $C_i(\chi_i)$ time units and all other tasks τ_j in τ generate jobs that run for at most $C_i(\chi_i)$ time units.

The procedure is repeatedly applied to the set of jobs excluding the lowest priority job, until all jobs are ordered, or at some iteration a lowest priority job does not exist.

Since the priority of the a task is based on its own criticality level, we can say that a task set is *Own Criticality Based Priority(OCBP)-schedulable* as long as we find a complete ordering of the tasks.

B. *Worst case end-to-end response time*

In real-time applications, *timeliness* is one of Quality of Service (QoS) parameters which has utmost importance. For example, in real-time video/audio streaming applications, delay matters. If the system is unable to deliver frames within a sliding window of period, the frames arrived outside the window will be discarded. So, this results in reduced Quality of Experience (QoE), i.e., less visual comfort for users. Shorter end-to-end response time helps the frames to arrive within sliding window and it contributes to the overall performance of the system.

Several approaches can be used to determine the maximum end-to-end response time, such as *stochastic* or *deterministic* approach [7]. Furthermore, the *deterministic* approach is divided into *holistic* and *trajectory* approaches. In our work, we consider *trajectory* approach that gives better estimation of the end-to-end response time when compared to the *holistic* approach.

C. JPEG2000

JPEG2000 is an image compression standard and coding scheme. In addition to its high coding efficiency, JPEG2000 also provides with a number of highly desirable features such as seamless progressive transmission by resolution or quality, lossy to loss-less compression, random codestream access and processing, and region of interest. In [8], JPEG2000 is covered in much detailed way.

D. AVAILABLE BANDWIDTH ESTIMATION

Accuracy of available estimated bandwidth and convergence delay algorithm are research challenges in wireless network measurements. In wireless communication networks, the available bandwidth could be used as an important parameter to take decisions concerning many issues such as load control, admission control and routing.

Available bandwidth estimation methods can be divided in two major approaches:

- *Intrusive approaches* - these methods are based on end-to-end probe packets to estimate the available bandwidth on the link.
- *Passive approaches* - they use local information on the used bandwidth (e.g. the channel usage computed by sensing the radio medium) and exchange this information through *Hello* messages that are used in many routing protocols.

In [9], Prasad et. al presented four types of bandwidth estimation tools which uses intrusive approaches. All techniques are provided in their paper.

In our work, we rely on an active probing available bandwidth estimation tool called *Wireless Bandwidth estimation tool (WBest)*. This algorithm is proposed in [10]. The authors demonstrated that *WBest* has higher accuracy and faster convergence time in wireless environment with respect to other tools. They made a comparison with existing available bandwidth estimation tools such as: IGI/PTR , PathChirp, and Pathload. They recommended *WBest* for multimedia streaming applications over wireless networks. In their work, they pointed out that finding the optimal length of the trains used in the steps is a difficult matter. They proposed that 10 packet pairs for the first train and 30 packets for the second train are good choices, which yield a sufficiently accurate bandwidth estimation results. We also adopt these choices in our work.

III. JPEG2000 Video Streaming

In this section, first we define our architecture for the wireless multimedia sensor network. The wireless network is composed of Raspberry Pi platforms[11] and we name it π-*sense* network. Then we setup the mixed-criticality scheduling model for the wireless network.

A. Network Setup

The raspberry pi platform is used as sensor node. It is a credit-card-sized single-board computer with several peripheries for different purposes. EDIMAX WIFI dongle [12] is attached to USB port of the raspberry pi board. The dongle complies with wireless 802.11 b/g/n standards with data rates up to 150 Mbps and supports smart transmit power control and auto-idle state adjustment.

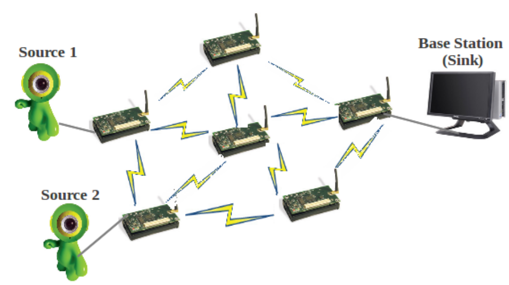

Figure 1. The π-sense wireless network

One remaining part to add to this section is the routing protocol we used in the π-sense network. In [13], Murray et al. made comparisons between *Optimized Link State Routing (OLSR)*, *Better Approach To Mobile Ad hoc Networking (BATMAN)*, and *Babel* routing protocols. They investigated the cause of performance loss or gain in multi hop ad hoc networks. They pointed out that in small networks, *Babel* offers higher throughputs due to reduced protocol overhead. In our mesh network of raspberry pi platforms, we use *Babel* routing protocol.

The whole network is depicted in figure 2. In the π-sense network, there are two sources transmitting video/image to the sink.

B. MC-wireless Model

In this section, we setup the proposed scheduling algorithm for the π-sense wireless network. Since the available bandwidth in the network varies, we took bandwidth estimates (bits/sec) in discrete format by using *WBest* estimation tool. We did this process offline so that we can know beforehand the bandwidth needed for the transmission of the images/frames.

We represent $B(l)$ as an estimate of available bandwidth at criticality level $l \in$ [1,L]. Furthermore, we assume that $B(l)$ is monotonically decreasing for increasing criticality level l, i.e.,

- $B(l+1) \leq B(l)\ \forall l \in [1, L]$

We consider mixed-criticality periodic frame sets $f = \{f_1, f_2, ..., f_n\}$ on non-preemptive communication channel where maximum criticality of the frame is L. The channel has bandwidth of $B(l)$ units per second, $\forall\ l \in$ [1,L].

Each MC frame is characterized by a 5 tuple $f_i = \{R_i, T_i, D_i, \chi_i, C_i\}$ where:

- $R_i \in \mathbb{N}$ is the release time of the first packet of the frame f_i,
- $T_i \in \mathbb{N}^*$ is the period of the frame f_i,
- $D_i \in \mathbb{N}^*$ is the deadline of the frame f_i, $D_i \leq T_i$,
- $\chi_i \in \mathbb{N}$ is the maximum criticality of the frame f_i, $\chi_i \leq L$,
- $C_i \in \mathbb{R}^+$ is an L vector of transmission times of frame f_i.

By definition, $C_i(l)$ is the time needed to transmit frame f_i of length N_i at criticality level $l \in [1, L]$. It is defined as:

$$C_i(l) = \frac{N_i}{B(l)}$$

The rest of this section deals with the computation of worst case end-to-end response time (WCERT). The WCERT is computed at each level of criticality.

*1) **Notations**:* The following notations are used in our work:

- $w_{i,m}$ is the relative start time of transmission of frame f_i for the m^{th} period T_i
- $WCERT_i$ is the worst case end-to-end response time of frame f_i
- $C_i^h(l)$ is transmission time of frame f_i from node h at criticality level l
- L_{max}, L_{min} - maximum/minimum network delay between two consecutive nodes in the wireless network
- Jin_i^1 is the maximum jitter of frame f_i at source node
- $hp(i,l)$ is the set of frames having a priority higher than f_i at criticality $\chi_i \geq l$
- $lp(i,l)$ is the set of frames having a priority lower than f_i at criticality $\chi_i \geq l$
- $H_i^{1,h}$ is the maximum delay incurred by frame f_i due to $f_j \in lp(i,l)$, while going from source node (1) to sink (h)

*2) **Worst case end-to-end response**:* To use *trajectory* approach, we assume that all flows from both sources follow the same path to the sink. In π-sense network, the sources are two hops away from the sink. Hence, nodes are marked from 1(source) to 3. The end-to-end response time is obtained by summing the delays incurred on each node along the path of the flows.

Due to non-preemption effect, the transmission of a high priority frame (from source 1) can be delayed so that the lower priority frame (from source 2) finishes its transmission. The maximum delay incurred by f_i due to $f_j \in lp(i,l)$ when both of them follow the same path to the sink can be given as:

$$\begin{cases} H_i^{1,1}(l) & = max_{f_j \in lp(i,l)} \left\{C_j^1(l) - 1\right\} \\ H_i^{1,h+1}(l) & \leq H_i^{1,1}(l) + max_{f_j \in lp(i,l)} \left\{C_j^{h+1}(l)\right\} \\ & -min_{f_j \in hp(i,l) \cup f_i} \left\{C_j^h(l)\right\} \\ & +L_{max} - L_{min} \end{cases}$$

Property 2 gives an upper bound on the maximum delay incurred on the path $h \in [source, sink]$. In [7], Martin et al. gave proof of the upper bound on the delay.

Before obtaining the worst case end-to-end response time, we need to find the latest release time of f_i when the path consists of q nodes. It is given by:

$$\begin{cases} w_{i,t}^q(l) & = \sum_{f_j \in hp(i,l)} (1 + \lfloor \frac{w_i^q(l) + Jin_i^1(l)}{T_j} \rfloor) \text{x} C_j^{slow}(l) \\ & +(1 + \lfloor \frac{t + Jin_i^1(l)}{T_i} \rfloor) \text{x} C_i^{slow}(l) \\ & + \sum_{h=1}^{q} (max_{f_j \in hp(i,l) \cup f_i} \left\{C_j^h(l)\right\}) - C_i^q(l) \\ & +H_i^{1,q}(l) + (q-1) \text{x} L_{max} \end{cases}$$

For the latest release time, the prove of existence of solution and the upper bound is also given in [7].

Finally, the worst case end-to-end response time is given by:

$$WCERT_i^{1,q}(l) = max_{k=0..K} \left\{w_{i,t}^q(l) + C_i^q(l) - k.T_i + Jin_i^1(l)\right\}$$

where the value of K is also provided in [7].

In the next section, implementation of MC-wireless is given. Experimental results are shown in section 5.

IV. MC-wireless Implementation

In this section, we implement mixed-criticality scheduling scheme for the π-sense WMSN. First, we assign fixed priority for the information transmitted from the two sources (as shown in figure 1). That is, at any time, source 1 gets higher priority over source 2. The priority assignment can be based on *location*, for example. Assume that source 1 is at the entrance of a building and source 2 is inside the building. It is necessary that frames from source 1 is transmitted to the sink even if the available bandwidth is dropped. In this situation, source 2 stops transmission so that source 1 transmits its frames within short period of time. When the available bandwidth is enough, both sources transmit their frames over the wireless network.

Secondly, we define criticality levels that corresponds to the values of the available bandwidth. In our work, we set number of criticality levels to 3, i.e., $l \in$ [1,3]. Hence, $B(l)$ is the available bandwidth at criticality level $l \in$ [1,3].

At source nodes, we implement JPEG2000 algorithm to encode the frames. The frames can be critical or non-critical depending on the information encoded in it. For instance, when the available bandwidth is too low, the frame is encoded with base layer and resolution (that means it is critical frame). Hence, even if the bandwidth is low, we can still get a frame with lower quality from source 1, but source 2 is disconnected due to its lower importance.

frame	frame length(KB)	BW needed(Mbps)
f_1	5.7	$B(l) \leq 1$
f_2	7.6	$1< B(l) \leq 1.8$
f_3	15.2	$B(l) > 1.8$

Table I
NEEDED BANDWIDTH FOR THE FRAMES

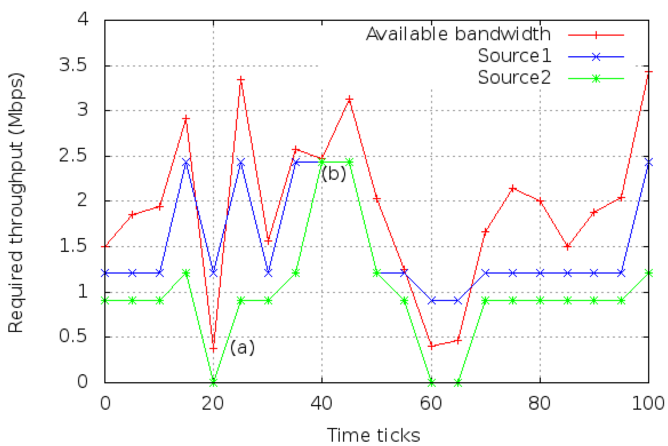

Figure 2. Required bandwidth for the sources

From table 1, we see that when the available bandwidth is greater than 1.8 Mbps, both critical and non-critical information (all layers with full resolution - i.e, f_3) can be transmitted to the sink. However, if the bandwidth is less than 1 Mbps, it is necessary to transmit only the critical information (i.e, f_1).

Figure 2 shows required bandwidth for the two sources along with estimated available bandwidth. The bandwidth values fall into range of [0.1, 3.427] Mbps. We divide the range into 3 parts. The first part is when the bandwidth is in [0.1, 1.0] Mbps. It corresponds to criticality level 3 (i.e., B(3) ∈ [0.1, 1.0]). Then, if the available bandwidth is in (1.0, 1.8] Mbps, we say that it is at criticality level 2 (i.e., B(2) ∈ (1.0, 1.8]). Finally, criticality level 1 corresponds to bandwidth values above 1.8 Mbps.

There are 2 important points to consider in figure 2. At point (a), the available bandwidth is dropped to 0.384 Mbps from prior value of 2.914 Mbps. This means, it is changed from criticality level 1 to 3 (i.e., the change is more than 1 step). In this situation, the appropriate selection from source 1 could be frame f_1 that requires a bandwidth of 0.912 Mbps (taking frame rate of 20 fps). However, due to comfort of user visualization, we select frame f_2 that requires 1.2 Mbps. Clearly, the time it takes to transmit f_1 is lower than that of f_2, but we have a good quality image at the end. Thus, it is a trade-off between the quality of the image and transmission time. Finally, it actually takes 158 ms (7.6KB/0.384Mbps) to transmit f_2. For source 2, we stop transmitting the frames because it has lower priority and the bandwidth is dropped too much.

Secondly, at point (b), the estimated available bandwidth is at criticality level 1. Hence , it is sufficient to transmit both critical and non-critical information from source 1. In this case, source 2 can also transmit its frames.

V. EXPERIMENTS

In this section, we present the results gained by adopting mixed-criticality scheduling to the π-sense network. The MC-wireless is compared against the classical case in which all information are considered equally important.

Let us consider a classical situation where all the information can be sent without considering the available bandwidth. This corresponds of transmission of both critical and non-critical information (i.e. f_3). In this case, even if the bandwidth is at criticality level 3, we transmit f_3. However, according to MC-wireless, we prioritize f_1 over f_3 because the the bandwidth required for f_3 is greater than that of f_1 and f_1 finishes transmission earlier than f_3. Hence, we get reduce transmission time by selecting f_1 over f_3.. This reduction in transmission time leads to an improved end-to-end response time.

The worst case end-to-end response time (WCERT) is shown in figure 3 for both classical and MC-wireless cases. From the figure, we see that MC-wireless improves the end-to-end response time due to the fact that MC classifies the information as critical/non-critical based on the available bandwidth. Experimental results of WCERTs are plotted for each criticality level, that is, WCERT_l represents the worst case end-to-end response time at criticality level $l \in [1, 3]$.

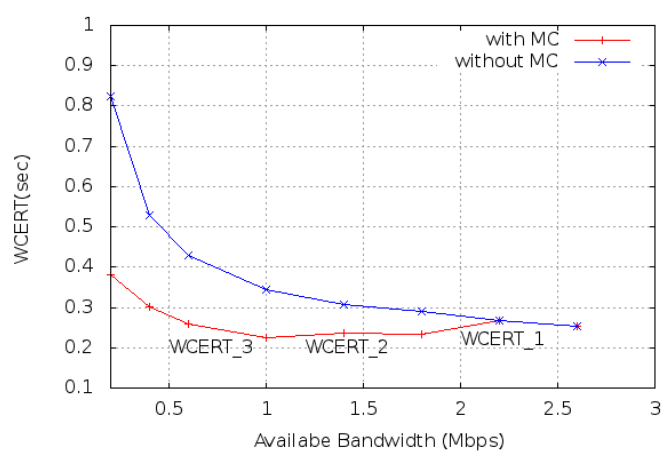

Figure 3. WCERT with and without MC

Let us consider a case when the available bandwidth is below 1 Mbps (region corresponding to WCERT_3). In this region, if we transmit all information (i.e. f_3), we end up incurring high WCERT. However, transmitting only the critical information (i.e. f_1), we gain a reduction in WCERT (from 0.822614 sec to 0.382091 sec when the available bandwidth is 0.2012 Mbps). This trend continues until the available bandwidth becomes more than 1.8 Mbps. Hence, in the third region ($WCERT_1$), since the available bandwidth is enough to send all the information, both MC-wireless and classical approaches provide the same value of WCERT.

To provide the upper bound on WCERT, we calculate end-to-end response time using the *trajectory* approach. On the path from source 1 to sink, we have 3 nodes. So, we compute delays at each node and obtain the response time for criticality level 3. Since source 2 is not transmitting at this criticality level, the delay due to non-preemption is ignored.

$$WCERT_1^{1,3}(3) = Jin_1^{source1}(3) + C_1^1(3) + C_1^2(3) + C_1^3(3) + 3*(L_{max} - L_{min}) = 0.623886 + \text{K sec}$$

where $Jin_1^{source1}(3)$ = 0.17023 sec, $C_1^1(3)$ = $C_1^2(3)$ = 5.7KB/0.2012Mbps = 0.2266 sec and $C_1^3(3)$ = 5.7KB/100Mbps = 0.000456 sec (the last node and sink are connected through Ethernet cable). We can assume that the processing time $(L_{max} - L_{min})$ on each node is not significant compared to other transmission times. Furthermore, since the bandwidth of every node is not exactly known, we can bound their bandwidths by estimated available bandwidth. Some of the nodes may have higher bandwidth, but estimating WCERT by available bandwidth will give an upper bound on the WCERT time. Hence,

$$WCERT_1^{1,3}(3) = 0.623886 \text{ sec.}$$

In our experiment, WCERT is found to be 0.382091 sec and it is 0.623886 sec using trajectory approach. The upper bound of WCERT will decrease if the capacity of each channel is known (as a case in LAN network).

VI. Conclusion and Future work

In this paper, we applied mixed-criticality scheduling scheme for wireless multimedia sensor networks. We showed the gains of adopting mixed-criticality in comparison to classical cases. An improved end-to-end response time is achieved by our experiments.

An interesting extension of our work can be to apply the proposed scheduling model to a larger network. In this case, it is possible to cluster nodes based on their location in the network. Such a scalable network will have a cluster head in each group. The cluster heads manage flows by using our proposed scheduling scheme. They also form another hierarchical layer and are linked to the sinks. Hence, applying the proposed scheduling scheme at each layer of the hierarchy allows addressing scalability issues in large networks.

References

[1] I.F. Akyildiz, T. Melodia, and K.R. Chowdhury, *Wireless Multimedia Sensor Networks: Applications and Testbeds*. IEEE Communications Surveys and Tutorials, VOL.10, NO.4, Fourth Quarter of 2008.

[2] S. Misra, M. Reisslein, and G. Xue, *A Survey of Multimedia Streaming in Wireless Sensor Networks*. Proceedings of the IEEE, VOL.96, NO.10, October 2008.

[3] S. Vestal, *Preemptive Scheduling of Multi-criticality Systems with Varying Degrees of Execution Time Assurance*. IEEE Computer Society, 2007, pp. 239-243.

[4] N.C. Audsley, *Optimal Priority Assignment and Feasibility of Static Priority Tasks with Arbitrary Start Times* . Tech. Rep YCS-164, 1991

[5] S. Baruah, V. Bonifaci, G. D'Angelo, H. Li, A. Marchetti-Spaccamela, N. Megow and L. Stougie, *Scheduling real-time mixed-criticality jobs*. IEEE Communications on Computers 61, 8(2012) 1140-1152

[6] N. Guan, P. Ekberg, M. Stigge and W. Yi, *Effective and Efficient Scheduling of Certifiable Mixed-Criticality Sporadic Task Systems*. IEEE Computer Society, 2011 pp. 183-192

[7] S. Martin, P. Minet, and L. George, *End-to-end response time with Fixed Priority Scheduling: Trajectory approach versus Holistic approach*. INRIA: France, 2004

[8] S. Taubman and W. Marcellin, *JPEG2000: Standard for Interactive Imaging* . Proceedings of the IEEE, VOL. 90, NO.8, August 2002.

[9] R.S. Prasad, M. Murray, C. Dovrolis, and K. Claffy, *Bandwidth Estimation: Metrics, measurement techniques and tools*. IEEE Network, vol 17, pp. 27-35, 2003

[10] M. Li, M. Claypool, and R. Kinicki *WBest: a Bandwidth Estimation Tool for IEEE 802.11 Wireless Networks*. Computer Science Department at Worcester Polytechnic Institute, Worcester, MA: USA

[11] Raspberry Pi : ⟨http://www.raspberrypi.org/⟩

[12] EDIMAX : ⟨http://www.edimax.com/en/produce_detail.php?pd_id=347&pl1_id=1⟩

[13] D. Murray, M. Dixon, and T. Koziniec, *An Experimental Comparison of Routing Protocols in Multi Hop Ad Hoc Networks*. Murdoch University: Australia, 2010