Selective Dropping Congestion Control for Wireless Multimedia Sensor Networks

A. Martelli, L. A. Grieco, M. Bacco, G. Boggia, P. Camarda
Dipartimento di Elettrotecnica ed Elettronica - Politecnico di Bari
Via E. Orabona, 4 - 70125 Bari (Italy)
e-mail: {a.martelli;a.grieco; m.bacco; g.boggia; camarda}@poliba.it

***Abstract*—A Selective Dropping Congestion Control (SDCC) algorithm is proposed in this work, for wireless multimedia sensor networks based on scalable multimedia encoding. SDCC, which has been designed by means of linear discrete time control theory arguments, can be applied in a fully distributed way. It periodically evaluates at each node the amount of data to drop, in order to control the level of the transmission queue, and then selects the packets to discard based on their relevance to the image quality. SDCC effectiveness has been evaluated with reference to a video-surveillance problem in several scenarios, using the *Castalia* simulator. Results have shown that SDCC is able to significantly improve the network performance in terms of image delivery ratio and delay.**

I. Introduction

Nowadays, technology is mature enough to allow the production of multimedia wireless sensors, i.e., wireless nodes able to acquire and process audio/video signals. A Wireless Multimedia Sensor Network (WMSN) relies on a set of collaborating multimedia wireless sensors to provide distributed multimedia sensing services. Potential WMSN application domains range over indoor/outdoor surveillance systems, traffic monitoring and control for urban and sub-urban areas, telemedicine, attendance to disable and elderly people, environment monitoring, localization and recognition of services and users, monitoring and control of manufacturing processes in industry [1]. The requirements of multimedia monitoring applications state new problems that the wireless communication and processing infrastructures have to solve to assure the desired Quality of Experience (QoE). Such problems include the limited power resources and computational capabilities, the computational complexity of multimedia compression, aggregation, and distributed processing, the overload to manage Quality of Service (QoS). With respect to classic Wireless Sensor Networks (WSNs), the constraints, imposed by the computational capacity, the memory and energy resources, become more critical. As a consequence, WMSNs require tools and methodologies different from the ones proposed for common WSNs [2].

The adoption of scalable encoding techniques for multimedia contents represents a very promising opportunity to increase WMSNs deployment in the real market [3]. In fact, these algorithms split the multimedia content over many bitstreams that can be mutually integrated or selectively dropped to increase or decrease data resolution, depending on the QoS provided by the communication infrastructure. In this way, when the available bandwidth is large enough, a high quality multimedia content is delivered, whereas, the quality of the transmitted signal is reduced in case of congestion and/or worsening of the wireless channel conditions. It is important to remark that to fully exploit the potentials of quality adaptive encoding techniques in WMSNs, each node has to dynamically select which packets have to be forwarded or discarded, depending on the locally available communication and computing resources.

It is proven that the most computationally efficient codecs, well suited to work on heavily constrained devices, are based on the wavelet transform [4]. In fact, through the multiresolution analysis, it allows the development of flexible techniques for bitstream scalability and encryption, such as those proposed in [5]-[7]. Among all wavelet-based encoders, the SPIHT codec has shown very good performances in terms of computational load and image quality, as it does not require entropy coding and it can be implemented in hardware [8]-[10]. It is one of the best candidates for wireless sensor networks applications [4] and many variants [11]-[13] exist on the original algorithm. We go one step further, using the FS-SPIHT [14] image codec, which keeps the main features of the SPIHT, such as the SNR scalability, and adds the *spatial scalability* without increasing the codec complexity. The bitstream produced by the codec is flexible and allows the nodes to optimally select the portion of the encoded image data to use, according to network conditions. Leveraging on the potentials of the FS-SPIHT codec, we designed a novel distributed congestion control algorithm for WMSNs, namely *Selective Dropping Congestion Control* (SDCC), which exploits a control theoretic approach to selectively discard buffered packets based on their relative importance with respect to the PSNR of the sensed images.

To the best of our knowledge, solutions already proposed in literature have a quite limited scope: the only contributions we found concern the use of resource intensive codecs like JPEG2000 [15] and H.264 [16], [17], or lightweight codecs like SPIHT [8]-[10] but with no support to spatial scalability.

SDCC has been designed by exploiting linear discrete time control theory to grant a limited computational complexity. According to the in-network processing paradigm, with SDCC, each node periodically evaluates the amount of data to drop in order to control the level of its transmission queue. To demonstrate its effectiveness in a relevant real-world scenario,

we have casted SDCC in a video-surveillance system, exploiting the very powerful Fully Scalable Set Partitioning In Hierarchical Trees (FS-SPIHT) image codec [14]. Simulation results obtained implementing SDCC in the *Castalia* simulator [18] have shown that it is able to greatly enhance WMSN performance under several networking conditions, improving both image delivery ratio and delay.

The rest of the paper is organized as follows: in Sec. II, we describe the FS-SPIHT algorithm. In Sec. III, the SDCC algorithm is proposed. Sec. IV shows simulation results. Finally, Sec. V draws the conclusions and states future research.

II. FS-SPIHT CODEC

In this Section we describe the most important features of the FS-SPIHT algorithm proposed in [14], focusing on those aspects that are very closely coupled with the scope of this work, which targets a general strategy for congestion control in WMSNs.

The FS-SPIHT algorithm enhances the capabilities of the SPIHT codec [19], by adding spatial scalability support. It exploits the spatial correlation among the subbands of a wavelet transformed image. This property allows to logically organize the wavelet coefficients in spatial orientation trees (see Fig. 1), in which each coefficient (except the top-left one in the upper level of the pyramid) has four children, each one belonging to a single subband. Since the coefficients in the topmost level of the pyramid are grouped in square groups of four, as shown in Fig. 1, the spatial orientation is simply the position of the coefficient in this square, and it can be identified as top-right, bottom-right or bottom-left. Inside a tree, the correlation between coefficients is very high, so that, if a coefficient is equal to zero, also its children are likely to be zero.

We chose the well known *Cohen-Daubechies-Feauveau 9/7* (CDF 9,7) wavelet, the same used in the JPEG2000 lossy profile, and we implemented it through *Lifting Scheme* [20], in order to reduce the computational complexity and to make it suitable for heavily constrained devices.

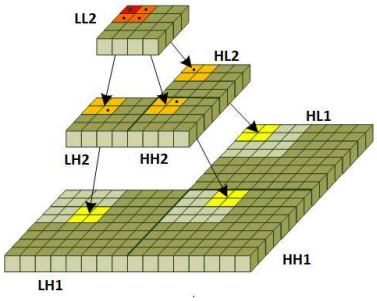


Fig. 1. Father-child dependencies in a spatial orientation tree

A N level wavelet transform generates $N+1 \equiv k_{max}$ resolution levels, going from 1 that represents the best spatial resolution (i.e., the full sized image) to k_{max}, indicating the worst spatial resolution.

The three subbands that have to be added to increase the resolution from level *k* to level *k-1* are referred to as level *k-1* resolution subbands. For instance, referring to Fig. 1, the image is decomposed in two resolution levels. To increase the resolution from level two to one, we need to add the subbands *LL1*, *HH1* and *HL1*.

Moreover, the algorithm allows the progressive transmission of the coefficients bits, starting from the most significant ones and ending with the least important ones, so that each more bit results in a refinement of the image quality. To this aim, FS-SPIHT encodes separately the different resolution levels in each bitplane, so allowing a parser or the decoder to directly access the data needed to reconstruct the image with the desired spatial resolution and/or distortion.

The bistream is arranged in a sequence of bitplanes ranging from n_{max} to 0, each one containing a sequence of resolution levels from k_{max} to 1. The n_{max} parameter represents the maximum number of bitplanes and depends on the magnitude of wavelet coefficients. It is defined by the formula

$$n_{max} = \left\lfloor log_2 \left(\max_{\{(i=1..h, j=1..w)\}} \{|c(i,j)|\} \right) \right\rfloor, \qquad (1)$$

where h and w are the height and the width of the transformed image. The k_{max} parameter represents the maximum number of resolution levels supported by the bitstream. Its maximum value is equal to the number of wavelet transform levels plus one.

Furthermore, the bitstream includes a general header, indicating the original image pixel size, n_{max} and k_{max}, and other small headers for every bitplane containing the offsets in byte of the resolution levels, which allow to efficiently parse the data.

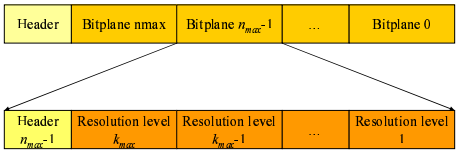


Fig. 2. FS-SPIHT Bitstream structure

For example, if we remove the resolution level 1 (the biggest one) in every bitplane, we can decode an half-sized version of the original image (spatial scalability). If, additionally, we remove some of the less significant bitplanes, we reduce the quality, and thus the PSNR, of the decoded image (SNR scalability).

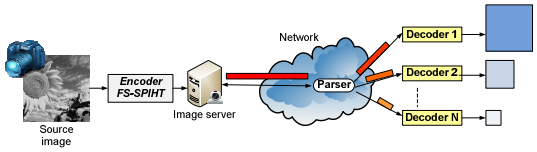


Fig. 3. Image generation and parsing in a FS-SPIHT enabled network

In our application, we decomposed the bitstream in a two-dimensional space (resolution level - bitplanes) and charac-

terized the image distortion (in terms of PSNR, computed as in [14]) for every possible decoding, considering an integer number of bitstream *segments* (Fig. 4). In what follows we will refer to *bitstream segment* as the information unit corresponding to one and only one spatial resolution level - bitplane pair.

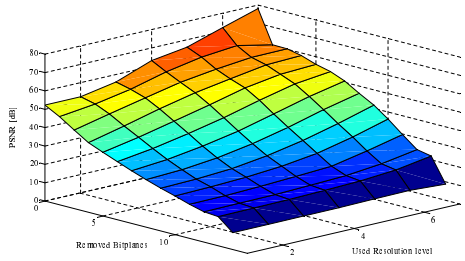


Fig. 4. Image distortion at various decoding precisions

Fig. 4 has been obtained analyzing the *lena* 512x512 grayscale image, but very similar results have been obtained also using other benchmark sequences. It displays the decoded image PSNR as a function of the number of used resolution levels and removed bitplanes: using only the highest resolution level without cutting any bitplane, we obtain the highest PSNR, but a very small image size. Furthermore, increasing the image size and the number of removed bitplanes, the PSNR gradually decreases.

Note that the algorithm requires the same space and computational resources as for the SPIHT, just with the addition of resolution dependent lists, and keeps producing a fully embedded bitstream with symmetric encoding and decoding times.

In what follows we will show how FS-SPHIT can support in-network selective data dropping in a WMSN to dynamically adapt the data stream to network conditions.

III. SDCC Algorithm

The network architecture we target in this work assumes a many-to-one convergent traffic going hop-by-hop in the upstream direction, i.e., from many sensors to the sink node [21]. Nodes are assumed to implement a CSMA-like MAC protocol [22]. Each sensor node can deal with two types of traffic: source and transit. The former is locally generated at the sensor node, whereas the latter comes from other nodes. Therefore, each sensor node can act as a source of data and/or as forwarding element. When the packet input rate at a node (given by source plus transit traffic rates) exceeds the packet forwarding rate at MAC layer (which depends on the MAC protocol itself, the radio channel, and the network topology) packets are stored in a First In First Out (FIFO) transmission buffer. Obviously, if no countermeasure is taken upon the saturation of the packet forwarding rate, the transmission queue builds up very fast until its maximum limit and packet losses arise due to congestion. Packet losses waste energy resources and can severely impair the network performances. Given that it is very difficult to regulate the packet forwarding rate, the reduction of the input traffic rate is the only possible action to face a network congestion. To face this problem, the SDCC algorithm acts in a fully distributed way by properly decreasing the number of queued packets at each node. Its rationale is to avoid buffer overflows by dropping some enqueued packets, chosen in relevance order, thus saving buffer space and energy resources. This is actually equivalent to virtually reduce the input rate. The total amount of data to drop is computed periodically by each sensor node using a control theoretic approach that drives the transmission queue toward a constant reference point. A schematic overview of the SDCC approach is pictured in Fig. 5: after each node selectively drops packets according to the congestion level in its neighborhood, the sink optimally merges received bitstream segments and decodes acquired images.

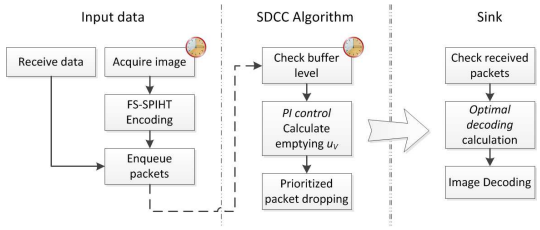


Fig. 5. Overall view of the SDCC algorithm

A. SDCC Control Law

Let $q(n)$ be the queue level associated with a transmission buffer at the sampling time n. In the discrete-time domain, its dynamics can be modeled by the following linear model [21]:

$$q(n+1) = q(n) + d(n) - u(n) + u_{_V}(n) \qquad (2)$$

where, with reference to the n-th sampling interval, $d(n) \geq 0$ is the number of packets enqueued, $u(n) \geq 0$ is the number of transmitted packets; $u_V(n) \leq 0$, computed by the SDCC algorithm, models the control action, i.e., its absolute value $|u_V(n)|$ is the number of packets that have to be removed from the queue using aggregation techniques.

More specifically, $d(n)$ is a function of the input traffic load at the queue, generated from the source nodes, and it may change with time; instead, $u(n)$ depends on the MAC protocol. Both $d(n)$ and $u(n)$ affect the queue size in a way that cannot be forecast in advance. For that reason, they will be modeled as disturbances in the sequel [21].

To avoid buffer overflows, SDCC computes, at each sampling time n, the amount $|u_V(n)|$ of packets that should be virtually removed from the queue.

In particular, SDCC is based on the feedback control loop pictured in Fig. 6. In such a scheme, the reference queue level q_T is compared against the actual queue level $q(n)$ and the resulting error $e(n) = q_T - q(n)$ is fed into a PI regulator with transfer function $G_c(z)$ to compute $u_V(n)$. We have chosen a PI controller because it is able to filter out the continuous component of the disturbance $d(n) - u(n)$ [21]. When the PI computes a positive value for $u_V(n)$, it is set

equal to 0. It is worth to note that $u_V(n)$ larger than zero means that the network is underloaded, and, as a consequence, there is no need to activate the aggregation algorithm.

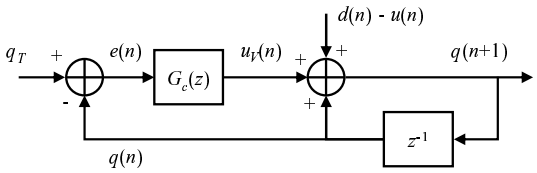


Fig. 6. Block Diagram of the controlled system.

The transfer function of the PI controller computed as the $\mathcal{Z}$-transform of its discrete-time pulse response, is given by

$$G_{_C}(z) = K \cdot \left(1 + \frac{1}{T}\frac{z}{z-1}\right) \tag{3}$$

where K and T are non negative constants. It can be equivalently expressed in the time domain using the following expression:

$$u_V(n) = u_V(n-1) + (K + K/T)e(n) - Ke(n-1) \tag{4}$$

To tune the PI regulator, the following theorem has been derived.

Theorem 1: The system shown in Fig. 6 is asymptotically stable if and only if the following inequalities are satisfied:

$$0 < K < 2; \qquad K/T > 0; \qquad 4 - 2K - K/T > 0. \tag{5}$$

Proof: This can be easily derived by applying the Jury criterion [23] to the characteristic polynomial $p(z) = T(z-1)^2 + K(T+1)z - KT$ of the control system in Fig. 6.

B. Packet Dropping Criteria

We map each bitstream segment in one network packet, identified by its bitplane and resolution level. If the segment size exceeds the maximum packet size, the segment is fragmented in more packets. Doing this, upon congestion, SDCC can scale the bitstream by removing an integer number of packets from the transmission buffers. The discarding priority of packets is set according to their relative importance with respect to the quality of the reconstructed image. In particular, packets having a smaller priority will be deleted first, according to the distortion characteristic in Fig. 4.

Following this criterion, a *zig-zag-like* packet scanning order allows to drop first those with the minimum impact on the PSNR, and then continue reducing simultaneously the image quality and its size. According to this rationale, $|u_V(n)|$ packets are searched and dropped at each iteration.

C. Optimal Image Reconstruction

Packets received by the sink are logically associated to a matrix M (see Fig. 7) with size $n_{BP}xn_{RL}$, where n_{BP} (n_{RL}) is the total number of bitplanes (resolution levels). The bottom left cell is referred to as $M(0,0)$.

As we can see in the example of Fig. 7, the FS-SPIHT decoder needs to select a *compact* (without missing segments) sub-rectangle having its bottom left corner coincident with the matrix one.

The *SDCC* algorithm aims at minimizing the probability of having holes in the middle of the matrix, but the possible absence of a bitstream segment causes the reconstruction rectangle to shrink. In this case, some theoretically useful packets are useless, and their transmission determines an energy inefficiency. [1]

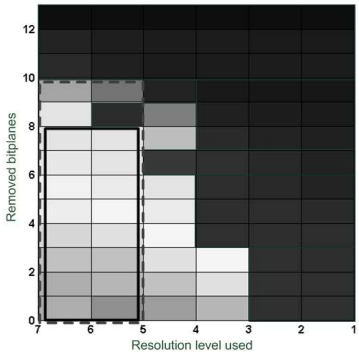


Fig. 7. Ex. of packets available at the sink. The dark ones are not present.

The *optimal decoding* consists in finding the biggest rectangle, measured in number of bits. We chose the number of bits as measure of the rectangle area, since the FS-SPIHT codec allows to slightly increase the image quality for every single bit added. The calculation of the optimal sub-rectangle is a search problem and can be easily and efficiently solved using a binary tree-like data structure and navigating it *breadth-first* [24].

IV. Performance Evaluation

To test the effectiveness of *SDCC*, we have simulated a WMSN for video-surveillance, using the *Castalia* simulator, v.2.3b [18]. Several *512x512* pixel grayscale images have been used, obtained as photographies of our Campus from several viewpoints. A CSMA/CA like MAC has been assumed with maximum packet size of 4096 bytes. Each node generates one frame per second.

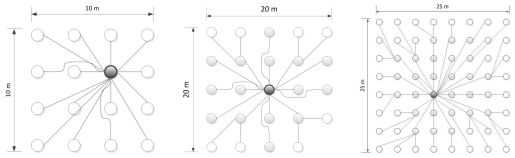
To grant system stability, the K parameter is constrained in the range $(0, 2)$ (see inequality (5)) whereas T can vary over a very broad set of values. For this reason, we set K to 1 and let vary T from 10 to 100.

We have considered three scenarios with increasing network loads, made by a set of sensors that deliver the acquired data towards a single sink located in the middle of the sensing area. We will measure several performance indices, such as: percentage, size, and PSNR of images delivered to the sink, as well as image delivery delay. Simulation parameter sets are reported in Tab. I. In each scenario the network nodes are disposed according to a regular square grid pattern. The number of nodes, as well as the field size and the average hop number, is increasing, as we can see in Fig. 8. As we

[1] Future enhancements could aim at optimizing the error resilience of the bitstream, i.e. the ability to reconstruct good quality images also with holes in the reconstruction rectangle.

TABLE I
PARAMETERS FOR SIMULATIONS

Parameter	*Description*	*Value*
BS	Buffer Size	8MB
q_T	Target queue level	90% BS
K	Constant of PI controller	1
T	Constant of PI controller	10:100
t_A	Sampling time of SDCC	90 ms
t_S	Duration of simulation	900 s
Rate	MAC Data Rate	55 Mbps
-	Number of nodes	16,24,64
-	Sensing area size	100, 400, 625 m^2



(a) 1-st scenario (b) 2-nd scenario (c) 3-rd scenario

Fig. 8. Simulated network topologies. The black node represents the sink.

will observe in the next sub-section, the number of nodes and the higher contention on the channel considerably affect the whole system performances, but using SDCC it is possible to mitigate these changes and to better control the quality of the delivered images.

A. Performance of SDCC

In this subsection we will show SDCC behaviour for $K = 1$ and $T = 90$ (a parameter set that grants for high performance gains on all the considered scenarios[2]). Tab. II shows that SDCC provides a great improvement in the system's performances. In fact, in all considered scenarios SDCC is able to delivery almost all acquired images to the sink (the image delivery percentage is above 99%). Furthermore, also the image delivery delay is improved, reaching values smaller than 4 s for the first two scenarios (the third one is very congested). This performance gain comes at the expense of the image size (and of the image PSNR in the first scenario), which is lowered by SDCC in order to fit the bandwidth constraints of the WMSN. In other terms, when SDCC is turned off larger images are delivered to the sink but with very high delays (not suitable for video surveillance applications) and loss ratios. On the opposite, using SDCC, smaller images are received by the sink with consequent benefits on the image loss ratio and delay. Different considerations have to be done for the PSNR. In fact, in the first scenario, SDCC provides a smaller PSNR, whereas, in the second and third ones, we observe a higher PSNR when SDCC is used. In order to explain this behavior, we have to remind that in the first scenario the network load is light so that, also if SDCC is not used, the network is able to deliver almost all the images. In other words, the packet loss ratio is small and, as a consequence, it is possible to reconstruct high quality images, even if this comes at the expense of a higher image delivery delay. On the opposite, in the second and third scenarios, the network load is higher, so that, the selective packet discarding mechanism of SDCC is able to drop only packets that carry information less relevant for the PSNR, thus enabling a performance gain.

[2]In the next sub-section we will evaluate parameter sensitivity.

TABLE II
SDCC PERFORMANCE ($K = 1; T = 90$)

	Scenario 1		Scenario 2		Scenario 3	
SDCC	OFF	ON	OFF	ON	OFF	ON
Received images [%]	97.30	100	85.05	100	75.87	99.10
Mean image size [pixel]	497×497	232×232	342×342	150×150	189×189	22×22
Mean PSNR [dB]	46.17	35.75	29.72	33.7	17.63	19.4
Mean delay $[s]$	49	0.5	96.88	3.88	196	67

B. Analysis of parameter sensitivity

Herein, we analyze the sensitivity of SDCC to the T parameter[3]. We observed that the image delivery percentage is not sensitive to T. On the other hand, Fig. 9 shows the image delivery delay as a function of T: in the first two scenarios, in order to let the delay falling below 5 s, the T parameter has to be sufficiently large. In the third scenario, instead, the load is so high the image delivery delay is always larger than 50 s. In order to explain the sensitivity to T, we have to consider that increasing T is equivalent to decreasing the integral action in the PI regulator (see Eq. (3)). This, in turn, makes the control signal u_V more sensitive to the instantaneous queue length than to its average. As a consequence, as T increases, the signal u_V becomes more and more bursty and SDCC starts dropping more and more packets from transmission queues, thus obtaining smaller images.

The delay improvement due to SDCC is mainly related to the mean image size (see Fig. 10), in fact, SDCC regulates the image size to the available bandwidth in order to mitigate network congestion. Scaling the image, reduces the amount of data to transmit and, as a consequence network induced delays. Since the packet dropping criteria contemporary regulates the image's pixel size and quality, it also affects the PSNR of the received images (see Fig. 11), which mainly depends on the network congestion.

V. CONCLUSIONS AND FUTURE RESEARCH

In this work, we have proposed an innovative distributed congestion control algorithm for WMSNs, which has been referred to as SDCC. The algorithm is able to select packets candidates to drop to face a network congestion. The effectiveness of SDCC has been evaluated with reference to a video-surveillance application, in three different scenarios, using the *Castalia* simulator and a flexible SNR and spatially scalable

[3]We did not report results obtained using different values of K because very similar to those we will show in what follows.

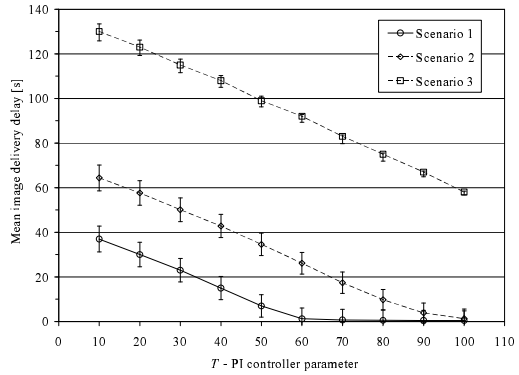


Fig. 9. Mean image delivery delay.

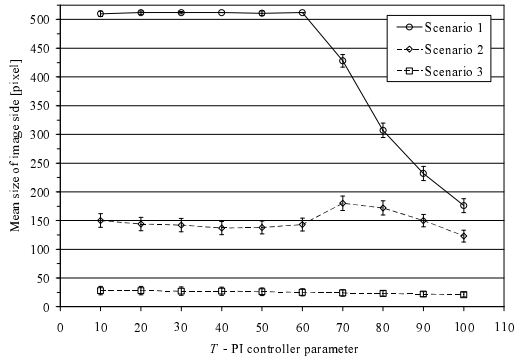


Fig. 10. Mean size of image side. A value n in the graph means that the image has size $n \times n$ pixels.

image codec (*FS-SPIHT*). Using this codec, we have also proposed algorithms for the bitstream scaling and the optimal image reconstruction at the sink. Results have shown that, using SDCC, network performance is significantly improved. Further research will consider: (i) codec optimizations related to security and error resilience; (ii) techniques to improve the energy efficiency of SDCC; (iii) comparison with existing congestion control algorithms for WSNs; (iv) evaluation of the impact of routing protocols on the networks overall performance; (v) adoption of image quality metrics other than PSNR.

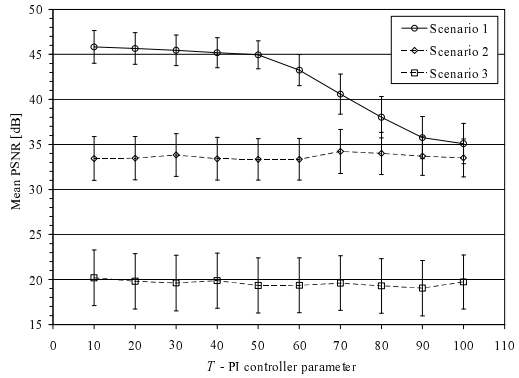


Fig. 11. Mean PSNR of images.

REFERENCES

[1] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks*, vol. 51, no. 7, pp. 790–802, July 2007.

[2] L. A. Grieco, G. Boggia, S. Sicari, and P. Colombo, "Secure wireless multimedia sensor networks: a survey," in *Proc. of UBICOMM*, Sliema, Malta, Oct. 2009.

[3] L. W. Chew, L. Ang, and K. P. Seng, "Survey of image compression algorithms in wireless sensor networks," *Int. Symp. on Information Technol.*, vol. 4, pp. 1–9, Aug. 2008.

[4] M. Wu and C. W. Chen, "Collaborative image coding and transmission over wireless sensor networks," *EURASIP Journal on Applied Signal Processing*, no. 1, pp. 223–223, Jan 2007.

[5] W. Wang, D. P. Honggang, and H. W. Sharif, "Image component transmissions in wireless sensor network," *IEEE Sarnoff Symp.*, pp. 1–5, April - May 2007.

[6] V. Lecuire, C. Duran-Faundez, and N. Krommenacker, "Energy-efficient transmission of wavelet-based images in wireless sensor networks," *Journal on Image and Video Processing*, vol. 7, pp. 15–15, 2007.

[7] C. ping Wu and C. c. Jay Kuo, "Design of integrated multimedia compression and encryption systems," *IEEE Trans. on Multimedia*, vol. 7, no. 5, pp. 828–839, Oct 2005.

[8] Zhi-Yan, C. Zheng-Zhou, and J. M.-Z. Hu, "An image sensor node for wireless sensor networks," *Int. Conf. on Information Technol.: Coding and Computing*, vol. 2, pp. 740–745, April 2005.

[9] W. Wang, D. Peng, and H. Wang, "Optimal image component transmissions in multirate wireless sensor networks," *IEEE Global Telecommunications Conf.*, pp. 976–980, Nov 2007.

[10] W. C. Chia, L. Ang, and K. P. Seng, "Multiview image compression for wireless multimedia sensor network using image stitching and spiht coding with ezw tree structure," *Int. Conf. on Intelligent Human-Machine Systems and Cybernetics*, vol. 2, no. 7, pp. 298–301, May 2009.

[11] M. Wu and C. W. Chen, "Multiple bitstream image transmission over wireless sensor networks," *Sensors, Proceedings of IEEE*, vol. 2, pp. 724–731, Oct 2003.

[12] Z. Jinchuang, T. Yan, and F. Wenli, "Research of image compression based on wireless visual sensor networks," *4th Conf. on Computer Science and Education*, pp. 353–356, Jul 2009.

[13] S. Rein, S. Lehmann, and C. Guhmann, "Wavelet image two-line coder for wireless sensor node with extremely little ram," *Data Compression Conf.*, pp. 252–261, Mar 2009.

[14] H. Danyali and A. Mertins, "Fully spatial and snr scalable, spiht-based image coding for transmission over heterogenous networks," *Journal of Telecommunications and Information Technol.*, vol. 2, pp. 92–98, 2003.

[15] W. Y. Sahinoglu, Z. Vetro, and A. Sahinoglu, "Energy efficient jpeg 2000 image transmission over wireless sensor networks," *Global Telecommunication Conf. GLOBECOM*, vol. 5, pp. 2738–2743, Nov-Dec 2004.

[16] D. Kandris, M. Tsagkaropoulos, I. Politis, A. Tzes, and S. Kotsopoulos, "A hybrid scheme for video transmission over wireless multimedia sensor networks," *17th Mediterranean Conf. on Control and Automation*, pp. 964–969, June 2009.

[17] Politis, I. Tsagkaropoulos, and S. M. Kotsopoulos, "Optimizing video transmission over wireles multimedia sensor networks," *IEEE Global Telecommunications Conf.*, pp. 1–6, Nov-Dec 2008.

[18] "Castalia simulator - official website," http://castalia.npc.nicta.com.au.

[19] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *Circuits and Systems for Video Technology, IEEE Trans. on circuits and systems for video technology*, vol. 6, pp. 243–250, 1996.

[20] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *Journal of Fourier Analysis and Applications*, vol. 4, no. 3, pp. 247–269, May 1998.

[21] T. Mastrocristino, G. Tesoriere, L. A. Grieco, G. Boggia, M. R. Palattella, and P. Camarda, "Congestion control based on data-aggregation for wireless sensor networks," in *Proc. of IEEE Int. Symp. on Industrial Electronics (ISIE2010)*, Jul. 2010.

[22] B. H. Walke, S. Mangold, and L. Berlemann, *IEEE 802 Wireless Systems*. John Wiley & Sons, 2006.

[23] K. J. Astrom and B. Wittenmark, *Computer controlled systems: theory and design*, 3rd ed. Prentice Hall, Englewood Cliffs, 1995.

[24] D. E. Knuth, *The art of computer programming*, 3rd ed. Addison-Wesley, 1997, vol. 1.