

Research Article

A Study on the Power Consumption of H.264/AVC-Based Video Sensor Network

Bambang A. B. Sarif,¹ Mahsa Pourazad,^{1,2} Panos Nasiopoulos,¹ and Victor C. M. Leung¹

¹Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, Canada V6T 1Z4

²TELUS Communications Inc., 555 Robson Street, Vancouver, BC, Canada V6B 3K9

Correspondence should be addressed to Bambang A. B. Sarif; bambangs@ece.ubc.ca

Received 25 March 2015; Accepted 26 August 2015

Academic Editor: Jong Hyuk Park

Copyright © 2015 Bambang A. B. Sarif et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

There is an increasing interest in using video sensor networks (VSNs) as an alternative to existing video monitoring/surveillance applications. Due to the limited amount of energy resources available in VSNs, power consumption efficiency is one of the most important design challenges in VSNs. Video encoding contributes to a significant portion of the overall power consumption at the VSN nodes. In this regard, the encoding parameter settings used at each node determine the coding complexity and bitrate of the video. This, in turn, determines the encoding and transmission power consumption of the node and the VSN overall. Therefore, in order to calculate the nodes' power consumption, we need to be able to estimate the coding complexity and bitrate of the video. In this paper, we modeled the coding complexity and bitrate of the H.264/AVC encoder, based on the encoding parameter settings used. We also propose a method to reduce the model estimation error for videos whose content changes within a specified period of time. We have conducted our experiments using a large video dataset captured from real-life applications in the analysis. Using the proposed model, we show how to estimate the VSN power consumption for a given topology.

1. Introduction

Technology advances in communications have enabled the implementation of pervasive computing applications that share the vision of small, inexpensive, distributed, and robust networked devices that can gather and process context specific information on behalf of the users. In this regard, wireless sensor networks (WSNs) [1] that can monitor different types of physical phenomena and are able to provide a diverse set of context data to interested clients can be used as the basis architecture for such implementation. While WSN was originally used to monitor physical measurement of the environment such as temperature and humidity, recent trends show that WSNs may successfully be used in a wide range of other applications, including monitoring the condition of public structures such as bridges [2], surveillance of access hatches [3], monitoring of indoor asbestos [4], healthcare [5], and habitat monitoring of seabird or fish [6, 7]. Furthermore, with the availability of more advance sensor nodes, we witnessed an increasing number of studies investigating the use

of sensor network platforms for intelligent environments [8], intelligent green service in the Internet of Things [9], and smart homes [10]. Some of these applications require the sensor network to provide multimodal information in the form of multimedia streams, such as images or video [11]. For this reason, video sensor networks (VSNs) have attracted a lot of research attention in the past decades. The low cost and flexibility offered by VSNs provide an interesting alternative to several existing video monitoring technologies [12, 13]. Studies on different VSNs applications have been reported in the literature [14–16].

Key research areas in VSNs are discussed in [11, 17], while [18] puts significant attention on sensor coverage, [19] details quality of service (QoS), and energy consumption is covered in [20–22]. However, since VSNs usually have limited energy resources, the issue of energy efficiency becomes one of the most important design aspects in VSNs. In a common WSN that operates on scalar data, energy efficiency is entirely dependent on the data transmission process [1, 23–25]. On the contrary, video processing requires extensive resources

in encoding the video and transmitting the encoded video stream. The encoder parameter settings used by the VSN nodes affect the coding complexity and bitrate of the video. The coding complexity and bitrate of the encoder in turn determine the encoding and transmission power consumption of the video node. In order to improve the VSN operation efficiency, an in-depth study of energy consumption trade-offs in a VSN is thus necessary.

Among the existing video coding standards, H.264/AVC is the most widely used video encoder in the consumer market [26, 27]. In the context of VSN, Ahmad et al. [21] studied the required energy for encoding and transmitting video content in the case of using H.264/AVC encoder. Unfortunately, the number of encoding configuration settings considered in that study is limited. By including more encoder settings than those used in [21], the authors in [22] proposed a table that includes different combinations of coding complexity and bitrates, producing compressed videos with almost similar quality in terms of peak signal to noise ratio (PSNR). A model to estimate the coding complexity and bitrate of an H.264/AVC-based VSN was proposed in [28].

In this paper, we modeled the coding complexity and bitrate of the H.264/AVC encoder in a VSN, based on the encoder parameter settings used. In order to proceed, we need to mimic a real-life setting of a VSN deployment and capture a large amount of real-life content which we used in our analysis. From this large dataset, some videos were used as the training set, while the rest were used to test the performance of our model. We provide a method to reduce the estimation error for videos whose content changes within a specified period of time. Using our proposed scheme, we show how the VSN total power consumption is estimated.

The rest of the paper is organized as follows. Section 2 describes the H.264/AVC coding complexity and bitrate modeling. The encoding and transmission power consumption model is discussed in Section 3. Conclusions are drawn in Section 4.

2. H.264/AVC Complexity and Bitrate Model

In this section we describe our coding complexity and bitrate model. A method for reducing the estimation error is also described in this section.

2.1. Experiment Settings. In order to mimic realistic VSN applications, we have captured real-life videos using four cameras in the atrium of a public building. The cameras were installed so that each of them had a different point of view as shown in Figure 1. The views of some cameras were overlapping with one another. The scene arrangement was such that each camera point of view was different. In order to mimic a practical application, all video sequences were downsampled to 416×240 pixels of resolution and their frame rate was reduced to 15 frames per second (fps). Five shots of videos were captured using the four cameras, resulting in a total of 20 different videos. These videos were named using the convention $\langle camera-id_shot-id \rangle$. Therefore, $camera1_shot1$ is the video obtained by camera 1 in the first

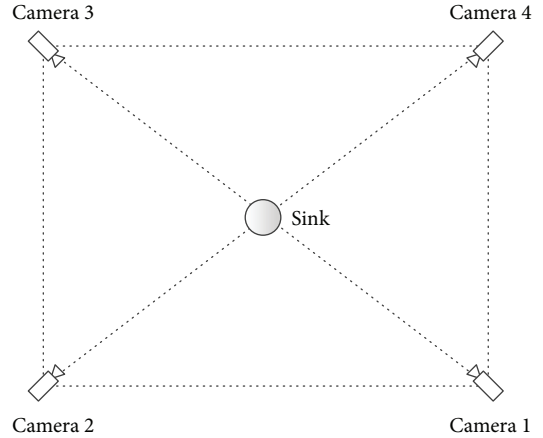


FIGURE 1: Camera placements.

shot. The four videos of the fifth shot were selected as the training set for the model, while the remaining videos were used as the test set.

In VSN applications, due to the limitations in energy and processing resources, less complex encoder configurations are used. To this end, we used the baseline profile of H.264/AVC that uses only I- and P-frames (no B-frames) and is suitable for low complexity applications. Note that, similar to its predecessor, H.264/AVC is a block based hybrid video encoder that utilizes intraframe and interframe prediction techniques. There are many parameters that control the encoding performance in terms of coding complexity and bitrate. The group of pictures (GOP) size that controls the number of interframe coded pictures in successive frames is a parameter that significantly affects the coding complexity and bitrate. The other factor that controls the complexity and the performance of the H.264/AVC codec is the number of block sizes used in the interprediction process. Increasing the number of block size candidates used in the interprediction results in a higher compression performance at the expense of increased complexity. In general, there are seven block sizes defined for interprediction in H.264/AVC. In this paper, the complexity of motion estimation (ME) is classified into different levels of complexity depending on the number of block size candidates used, as shown in Table 1.

The H.264/AVC reference software JM version 18.2 is used in our experiments. In addition to using only I- and P-frames, we also used context adaptive variable-length entropy coding (CAVLC) and one reference frame. Other settings include search range (SR) for motion estimation equal to 8, disabling the rate distortion optimization (RDO), rate control, subpel motion estimation, deblocking filter, and intracoding for P-frames options. The quantization parameter (QP) used to encode all videos is set equal to 28. Furthermore, to have an objective measure for the coding complexity, we use the number of basic instructions count to encode the video. This is provided by the instruction level profiler *iprof* [29]. We developed the coding complexity and bitrate models by considering the effect of GOP size and the number of block size candidates used to encode the video. These models

TABLE 1: ME complexity level (M_L) and δ_{M_L} .

M_L	Block size candidates	δ_{M_L}
1	SKIP, 16×16	0
2	SKIP, $16 \times 16, 16 \times 8$	0.13
3	SKIP, $16 \times 16, 16 \times 8, 8 \times 16$	0.26
4	SKIP, $16 \times 16, 16 \times 8, 8 \times 16, 8 \times 8$	0.54
5	SKIP, $16 \times 16, 16 \times 8, 8 \times 16, 8 \times 8, 8 \times 4$	0.67
6	SKIP, $16 \times 16, 16 \times 8, 8 \times 16, 8 \times 8, 8 \times 4, 4 \times 8$	0.81
7	SKIP, $16 \times 16, 16 \times 8, 8 \times 16, 8 \times 8, 8 \times 4, 4 \times 8, 4 \times 4$	1

are explained in detail in our previous work in [28]. The following subsections provide the basic information about the modeling process.

2.2. Coding Complexity Modeling. The coding complexity of a video sequence (C_S) is formulated as follows:

$$C_S = C_I \cdot n_I + C_P \cdot n_P, \quad (1)$$

where C_I is the average coding complexity to encode an I-frame, C_P is the average coding complexity to encode a P-frame, n_I is the number of I-frames in the sequence, and n_P is the number of P-frames in the sequence. For a video sequence with no scene change, the value of C_I can be considered almost constant. On the other hand, C_P depends on the complexity level of the ME process. From our previous study, we noticed that the GOP size does not affect the normalized coding complexity of P-frames for any ME complexity level (M_L). In fact, using some scaling and normalization, δ_{M_L} can be defined as the fractional increase of normalized C_P at different M_L (see Table 1) [28]. The encoding complexity of a P-frame is then calculated as

$$C_{P_{M_L=i}} = C_{P_{M_L=1}} \cdot (1 + \delta_{M_L}(i) \cdot \omega_1), \quad (2)$$

where ω_1 denotes the range of normalized C_P values for a specific video. The value of ω_1 is calculated using a simple linear formula from the training videos. In this paper, the range of normalized C_P is modeled as follows:

$$\omega_1 = a \cdot C_{P_{M_L=1}} + b, \quad (3)$$

where $C_{P_{M_L=1}}$ is the average coding complexity to encode a P-frame using $M_L = 1$ and a and b are obtained using the least square regression technique on the training video data. Considering that $n_I = N/\text{GOP}$, where N is the total number of frames and $n_P = N - N/\text{GOP}$, the average complexity per frame is then computed as follows:

$$C_f = \frac{(C_I + C_{P_{M_L=1}} \cdot (1 + \delta_{M_L} \cdot \omega_1)) \cdot (\text{GOP} - 1)}{\text{GOP}}. \quad (4)$$

2.3. Bitrate Modeling. Similar to the coding complexity model, the total size of the encoded video sequence (in bits) is modeled as

$$R_S = R_I \cdot n_I + R_P \cdot n_P, \quad (5)$$

where R_I is the average size of an I-frame and R_P is the average size of a P-frame. The value of R_P depends on M_L and GOP used by the encoder. R_P is modeled as follows:

$$R_P = R_{P_{M_L=1}} \cdot (f(M_L) + f(\text{GOP})). \quad (6)$$

Here, $f(M_L)$ is a decay function with respect to M_L , which is modeled using the generalized logistic function. On the other hand, $f(\text{GOP})$ is modeled using $\omega_2 \cdot \ln(\text{GOP})$ [28]. In order to obtain the parameters for $f(M_L)$, we use the least mean square curve fitting of the normalized R_P of the training video sequences when $\text{GOP} = 2$. Using $f(M_L)$ obtained in [28], the average bitrate of a frame (R_f) is then estimated as

$$R_f = \frac{R_I}{\text{GOP}} + R_{P_{M_L=1}} \cdot \left(\left(p + \frac{q-p}{1 + e^{-r(\delta_{M_L}-s)}} \right) \cdot \frac{(\text{GOP} - 1)}{\text{GOP}} + \omega_2 \cdot \ln(\text{GOP}) \cdot \frac{(\text{GOP} - 1)}{\text{GOP}} \right), \quad (7)$$

where $R_{P_{M_L=1}}$ is the bitrate of P-frame when $\text{GOP} = 2$ and $M_L = 1$ and ω_2 is the weight for $f(\text{GOP})$. The value of ω_2 was estimated using least square regression of the training sequences.

2.4. Implementation of Our Model. In order to implement our model using the complexity and bitrate modeling, we need to obtain several variables from each video sequence. To this end, we encode the first two frames of each video sequence. For the bitrate model, R_I is assumed to be equal to the bitrate of the encoded first frame, while $R_{P_{M_L=1}}$ is equal to the bitrate of the second frame. In addition, the parameters for (7) used in this paper are as follows: $p = 0.92$, $q = 1$, $r = -21.36$, and $s = 0.14$ [28].

For the complexity modeling, the *ipprof* tool will provide us with the complexity of encoding the first two frames of the video sequence; that is, $C_{2\text{-frames}} = C_I + C_{P_{M_L=1}}$. In order to obtain the value of $C_{P_{M_L=1}}$ we need to estimate the value of C_I . We assume that, for the I-frame, the value of C_I can be estimated from the value of R_I using a linear regression of the training videos [28]. Thus, C_I is estimated using the formula $C_I = 0.0637 \cdot R_I + 214.56$ in this paper. Furthermore, the value of ω_1 is calculated using (3) and the following parameters: $a = 0.0135$ and $b = -2.13$.

2.5. Proposed Method to Reduce the Estimation Error. In many real-life captured videos, content may change during a 10 s video shot. For example, Figure 2 shows frames 1, 70, and 100 of the *camera1_shot3* video sequence. It can be seen that the content at the start of the video (frame number 1) differs significantly from the content towards the end of the video (frame number 100). On the other hand, Figure 3 shows frames 1, 60, and 110 of the *camera2_shot2* video sequence, where the content at the start differs significantly from the ones captured at a later time, that is, frames 60 and 110. Looking at the two figures, it is clear that obtaining the model parameters from the first two frames at the beginning of

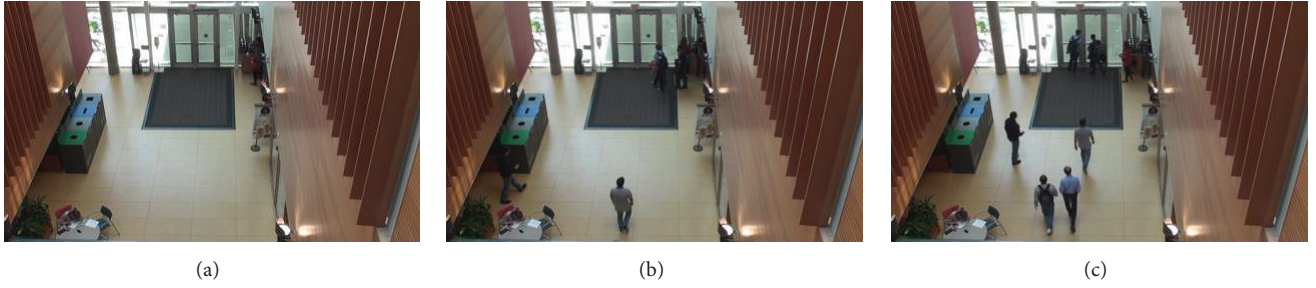


FIGURE 2: Content changes during a 10 s *camera1_shot3* video sequence. (a) Frame 1; (b) frame 70; (c) frame 100.

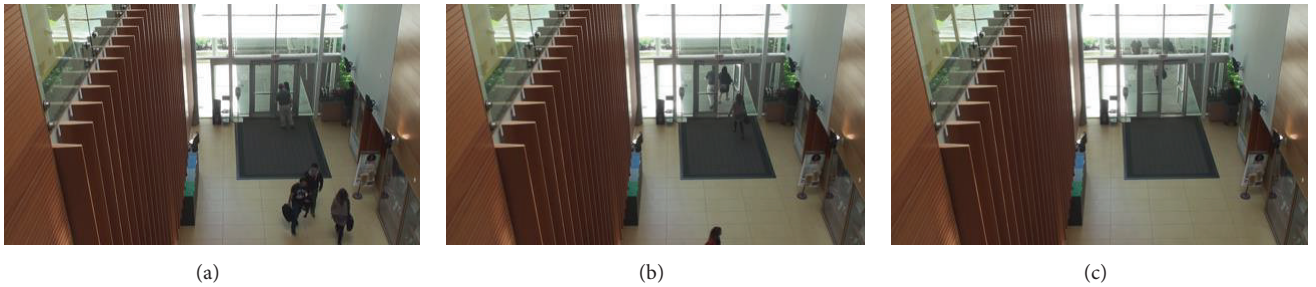


FIGURE 3: Content changes during a 10 s *camera2_shot2* video sequence. (a) Frame 1; (b) frame 60; (c) frame 110.

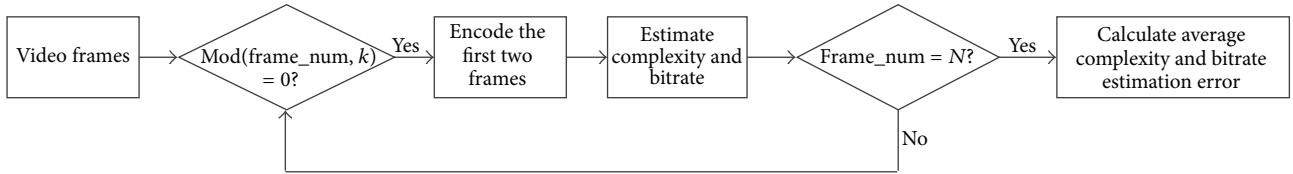


FIGURE 4: Flowchart for complexity and bitrate estimation error calculation.

the video may lead to a large estimation error. In order to tackle this problem, we divide the 10 s video into a number of subshots. In each subshot, bitrate and coding complexity estimation are performed. Figure 4 shows the flowchart of the proposed method to reduce the coding complexity and bitrate estimation error used in this paper. In that figure, the variable *frame_num* is the current frame number, while *k* denotes the length of a subshot in terms of the number of frames. Note that since the video is divided into $\lceil N/k \rceil$ subshots, the first two frames of each subshot are encoded to obtain the required parameters for the model.

In [30], the estimation error is calculated as the average estimation error of all the subshots. However, in order to provide a fair comparison, we calculate the estimation error from the complexity per second (C_{ps}) and average bitrate (R_{av}), defined as follows:

$$C_{ps} = \frac{F_r}{N} \sum_{i=1}^{\lceil N/k \rceil} C_f^i \cdot K_i, \quad (8)$$

$$R_{av} = \frac{F_r}{N} \sum_{i=1}^{\lceil N/k \rceil} R_f^i \cdot K_i.$$

Here, F_r is the frame rate, while K_i is calculated as follows:

$$K_i = \begin{cases} k, & 1 \leq i \leq \lfloor \frac{N}{k} \rfloor, \\ \text{mod}(N, k), & \text{otherwise.} \end{cases} \quad (9)$$

2.6. Analysis of the Model. In order to estimate the modeling error, the root mean square error (RMSE) of the coding complexity and bitrate for $\text{GOP} = \{1, 2, 4, 8, 16, 32, 64\}$, $M_L = \{1, 2, 3, 4, 5, 6, 7\}$, and $k = \{150, 75, 60, 45\}$ are calculated. The test set (TS) consists of the 16 videos shown in Table 2.

Table 3 shows the coding complexity estimation error of all test sequences and different values of *k*. The table shows that, in general, the coding complexity estimation error decreases as we use a larger number of subshots, that is, using smaller *k* values. We can also see that the proposed method manages to reduce the coding complexity estimation error in 11 out of 16 cases when *k* is set equal to 45. On the other hand, using *k* = 60 frames, the coding complexity estimation error is reduced in 13 out of 16 cases. In particular, in the case of video TS9, the coding complexity estimation error for *k* = 150 and *k* = 60 is equal to 69.666 and 37.266, respectively. This is equal to 46.5% reduction in estimation error. Figure 5

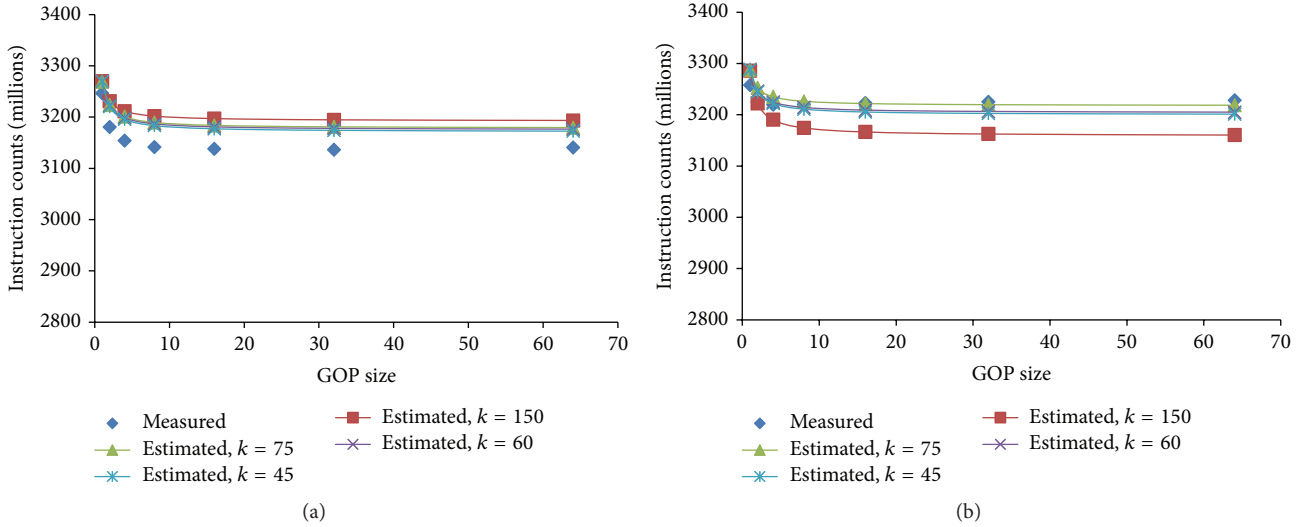


FIGURE 5: Measured and estimated coding complexity for different values of k and GOP sizes for video sequences (a) TS4 and (b) TS9.

TABLE 2: Test sequences.

Test sequence	Video name
TS1	camera1_shot1
TS2	camera2_shot1
TS3	camera3_shot1
TS4	camera4_shot1
TS5	camera1_shot2
TS6	camera2_shot2
TS7	camera3_shot2
TS8	camera4_shot2
TS9	camera1_shot3
TS10	camera2_shot3
TS11	camera3_shot3
TS12	camera4_shot3
TS13	camera1_shot4
TS14	camera2_shot4
TS15	camera3_shot4
TS16	camera4_shot4

TABLE 3: Coding complexity estimation error for different values of k .

Test sequence	$k = 150$	$k = 75$	$k = 60$	$k = 45$
TS1	34.966	35.533	28.523	27.890
TS2	26.722	26.812	26.823	30.280
TS3	48.005	38.997	45.258	35.861
TS4	45.437	36.435	34.667	32.790
TS5	33.850	32.615	37.589	34.662
TS6	37.247	29.967	26.934	26.985
TS7	28.769	36.088	28.459	32.256
TS8	33.145	27.086	27.538	27.052
TS9	69.666	30.149	37.266	39.555
TS10	59.759	29.830	39.279	30.596
TS11	47.022	37.739	35.961	39.236
TS12	41.304	33.905	35.581	31.479
TS13	27.858	32.782	30.906	32.127
TS14	38.642	38.363	31.962	33.426
TS15	36.970	36.930	36.914	36.860
TS16	39.797	39.986	32.818	39.890

shows the plot of the measured coding complexity and the estimated coding complexity per second (C_{ps}) for different values of k and varying GOP sizes. Note that, in this figure, the value of M_L is set to four.

Furthermore, Table 4 shows the bitrate estimation of all test sequences and different values of k . Similar to the coding complexity case, the table shows that, in general, the bitrate estimation error decreases as we use smaller k values. We can also see that the proposed method manages to reduce the bitrate estimation error in 12 out of 16 cases when k is set equal to 45. However, when k is set equal to 60, the bitrate estimation error is reduced in 13 out of 16 cases. The highest error reduction is obtained in the case of the TS9 video sequence. In this particular video, the RMSE of the bitrate model for $k = 150$ is equal to 75.219 kbps. However, when

k is set equal to 60 frames, the RMSE of the bitrate model is reduced to 33.851 kbps. This is equal to 55.7% reduction in the estimation error. Figure 6 shows the plot of the measured bitrate and the estimated average bitrate (R_{av}) for different values of k and varying GOP sizes. Note that, in this figure, the value of M_L is set equal to four.

The results analyzed in the previous paragraphs show that, by dividing the video sequences into a number of subshots, the model estimation error is reduced. The results also show that the reduction of the estimation error varies from one video to another. However, it is observed that setting $k = 60$ provides us with the smallest estimation error. From this point onward, the power consumption analysis of the VSN is performed under the assumption that the value of k is set equal to 60 frames.

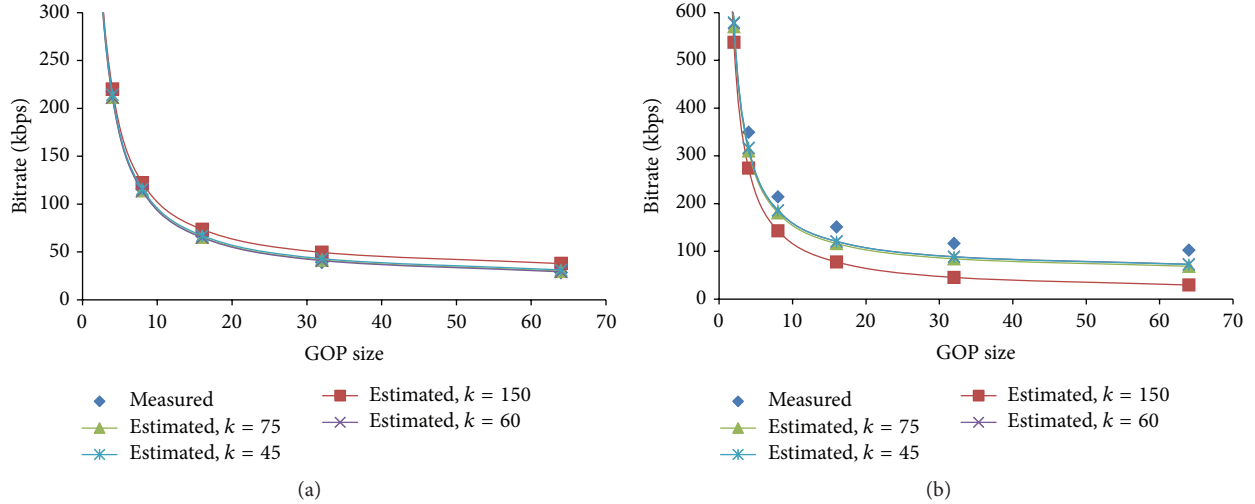


FIGURE 6: Measured and estimated bitrate for different values of k and GOP sizes for the video sequences (a) TS4 and (b) TS9.

TABLE 4: Bitrate estimation error for different values of k .

Test sequence	$k = 150$	$k = 75$	$k = 60$	$k = 45$
TS1	34.518	25.418	25.172	13.673
TS2	19.536	15.413	15.931	7.517
TS3	16.299	8.806	9.189	8.517
TS4	6.920	6.256	6.201	4.397
TS5	8.105	4.326	10.246	8.994
TS6	1.729	3.986	5.002	8.189
TS7	16.081	14.566	13.806	12.671
TS8	23.886	5.123	11.233	11.414
TS9	75.219	39.312	33.851	33.323
TS10	46.566	27.678	28.663	19.118
TS11	11.960	9.755	10.143	9.213
TS12	12.459	5.488	1.949	3.270
TS13	10.359	15.375	20.977	19.560
TS14	18.760	22.383	17.658	15.693
TS15	9.920	9.854	9.747	9.843
TS16	6.576	7.030	6.678	7.041

3. Power Consumption Estimation and Analysis

The power consumption of a video node in a VSN consists of encoding energy consumption and communication power consumption. The power consumption for encoding is estimated as follows:

$$P_e = C_{ps} \cdot CPI \cdot E_c, \quad (10)$$

where CPI is the number of CPU cycles to perform one basic instruction and E_c is the energy depletion per cycle. On the other hand, the transmission power consumption is calculated as

$$P_t = \sum (\alpha + \beta \cdot d^n) \cdot R_{av}, \quad (11)$$

where α is a constant coefficient related to coding and modulation, β is the amplifier energy coefficient, d is the transmission distance, and η is path loss exponent.

For our analysis, we use the topology shown in Figure 1, consisting of four video nodes and a sink. The parameters shown in Table 5 are used for the experiments. In order to analyze the effect of different video sources and encoding configurations, two sets of experiments are conducted. In the first experiment, the nodes' encoder parameter settings are set to be the same in all scenarios. However, the video sources used in each scenario vary. On the other hand, in the second experiment, the nodes are configured to use the same set of video sources in all scenarios, while the nodes' encoding parameter settings and the nodes' distance to the sink are varied.

The scenarios' configuration for the first experiment is shown in Table 6. In the first scenario, the VSN nodes are using the videos obtained from the first shot: *camera1_shot1*, *camera2_shot1*, *camera3_shot1*, and *camera4_shot1*. On the other hand, in the second scenario of the first experiment, the videos used are the videos obtained from the second shot and so on. Note that, for this experiment, M_L value is set equal to six. Figure 7 shows the estimated nodes' power consumption in the first experiment. The figure shows that the nodes' power consumption in each scenario is not the same. We can also observe that the trend of nodes' power consumption profile for each scenario varies. For example, Figure 7(a) that corresponds to scenario 1 shows that the node that consumes the highest power consumption is node 3. However, the difference in terms of total power consumption between node 3 and the other nodes in this scenario is not significant. In the other scenarios (i.e., scenario 2, scenario 3, and scenario 4), however, the node that has the highest power consumption is node 1. It is interesting to see that the encoding power consumption of each node in each scenario is not the same, even though all nodes are using the same encoding parameter settings in this experiment. In addition, the variance of nodes' total power consumption varies between one scenario and the other. In terms of VSN's average power consumption, we

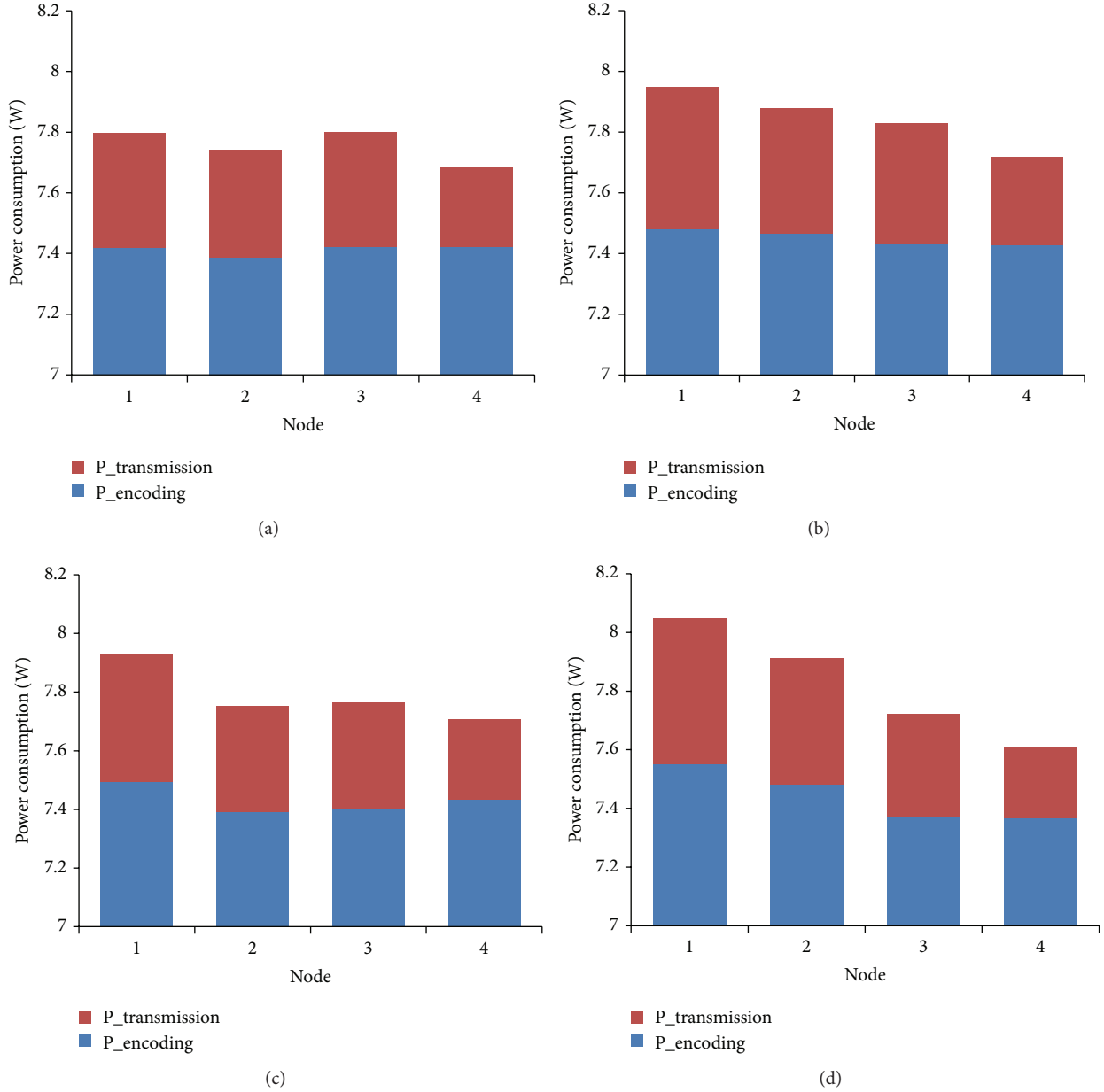


FIGURE 7: Nodes' power consumption in experiment 1: (a) scenario 1, (b) scenario 2, (c) scenario 3, and (d) scenario 4.

obtained the following values: 7.756 W (scenario 1), 7.843 W (scenario 2), 7.787 W (scenario 3), and 7.824 W (scenario 4). These results show that the content captured by each camera node affects not only the node's power consumption but also the VSN's average power consumption.

In the second set of experiments, the VSN nodes are set to use the videos from the first shot. However, the nodes' distance to the sink and the GOP size are varied. M_L is set equal to six, similar to the first experiment. The configuration used in the second experiment is summarized in Table 7. Figure 8 shows the estimated nodes' power consumption in this experiment. Figures 8(a) and 8(b) show the nodes' power

TABLE 5: Parameters used.

Symbol	Definition	Value
F_r	Frame rate	15 fps
N	Number of frames	150 frames
k	The length of subshot	60 frames
CPI	Average cycle per instruction	1.78
E_c	Energy consumption per cycle	$1.215e - 9$ J/cycle
α	Energy cost for transmitting 1 bit	$1e - 9$ J/b/m ⁴
β	Transmit amplifier coefficient	$5e - 8$ J/b
η	Path loss exponent	3.5

TABLE 6: Experiment 1 scenarios.

Scenario	Test sequences used	Distance to the sink	GOP size
1	TS1 (node 1), TS2 (node 2), TS3 (node 3), TS4 (node 4)	3 m	8
2	TS5 (node 1), TS6 (node 2), TS7 (node 3), TS8 (node 4)	3 m	8
3	TS9 (node 1), TS10 (node 2), TS11 (node 3), TS12 (node 4)	3 m	8
4	TS13 (node 1), TS14 (node 2), TS15 (node 3), TS16 (node 4)	3 m	8

TABLE 7: Experiment 2 scenarios.

Scenario	Test sequences used	Distance to the sink	GOP size
1	TS1 (node 1), TS2 (node 2), TS3 (node 3), TS4 (node 4)	1.5 m	2
2	TS1 (node 1), TS2 (node 2), TS3 (node 3), TS4 (node 4)	1.5 m	16
3	TS1 (node 1), TS2 (node 2), TS3 (node 3), TS4 (node 4)	5 m	2
4	TS1 (node 1), TS2 (node 2), TS3 (node 3), TS4 (node 4)	5 m	16

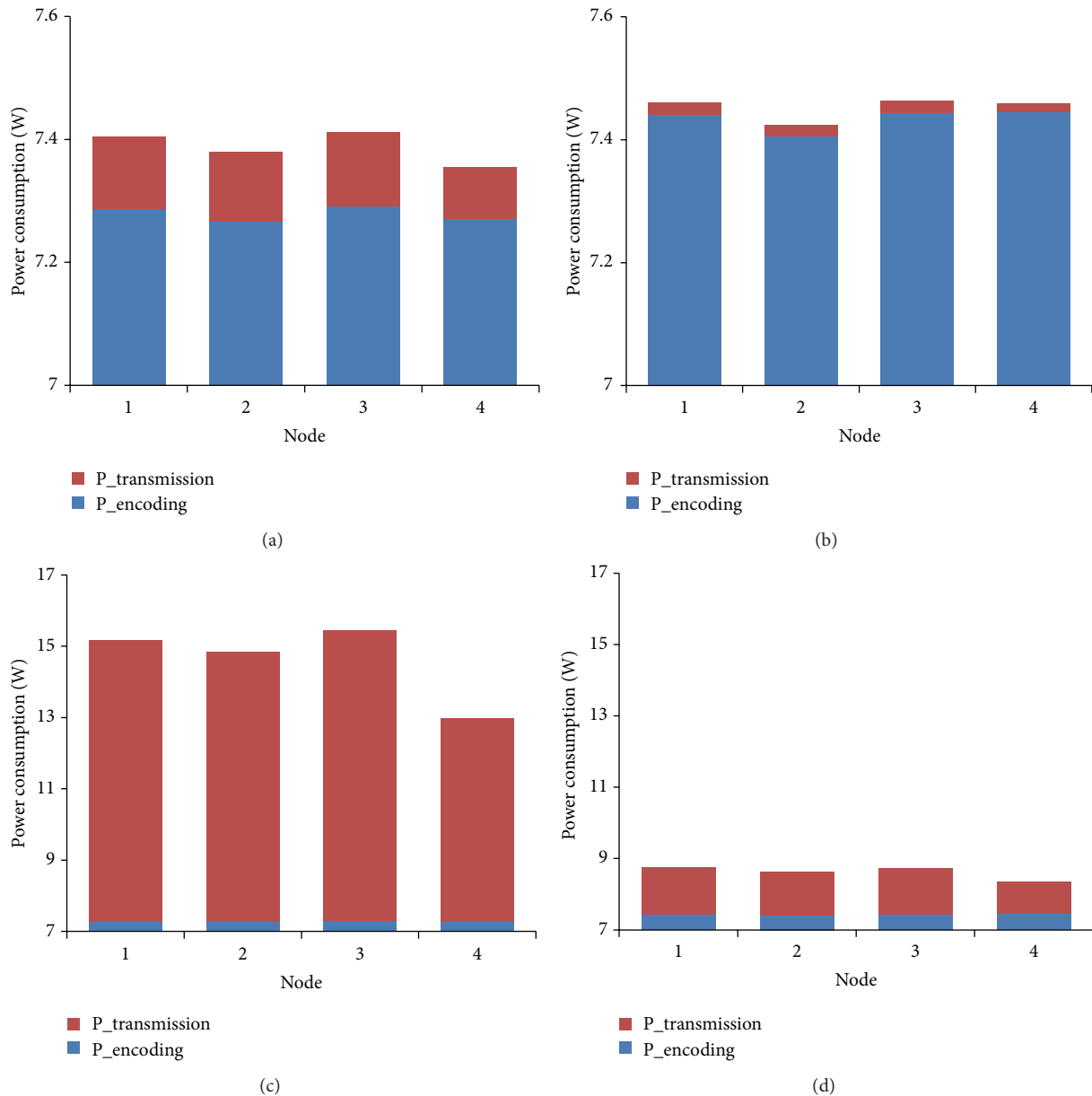


FIGURE 8: Nodes' power consumption in experiment 2: (a) scenario 1, (b) scenario 2, (c) scenario 3, and (d) scenario 4.

consumption when the nodes' distance to the sink is equal to 1.5 m for scenario 1 and scenario 2, respectively. However, the GOP size is set equal to 2 (scenario 1) and 16 (scenario 2). It can be seen from these figures that when the distance to the sink is small, using smaller GOP size will reduce the node's power consumption. The VSN's average power consumption shown by these figures is 7.388 W (scenario 1, GOP = 2) and 7.451 W (scenario 2, GOP = 16). The node's power consumption can be further reduced if the nodes are configured to use GOP equal to one. In this case, the VSN's average power consumption will be equal to 7.316 W. Furthermore, Figures 8(c) and 8(d) show the nodes' power consumption when d is equal to 5 m. Similar to the previous case, the GOP size is set equal to 2 and 16 for scenario 3 and scenario 4, respectively. It can be seen clearly from these figures that the cost of transmitting the encoded video increased tremendously as compared with the first two scenarios when d is smaller. Therefore, when the nodes' distance from the sink is large, the node's power consumption can be reduced if bigger GOP sizes are used. Comparing Figures 8(c) and 8(d) we observe that the VSN's average power consumption for these scenarios is 14.607 W (scenario 3) and 8.616 W (scenario 4), respectively. These results show that the node's VSN power consumption depends on the encoding configuration used and the distance between the node and the sink.

4. Conclusion

In this paper, we have proposed a new scheme for estimating the VSN power consumption. The scheme is based on using a coding complexity and bitrate model that incorporates some important encoding parameter settings. Through an adaptive scheme for adjusting the model parameters, we showed that the model estimation error could be reduced. Using our model, we analyzed the VSN node's power consumption under different scenarios that involved the use of various video content, encoding configurations, and nodes' distance from the sink. We showed that the VSN nodes' power consumption depends on the encoding parameter settings, the complexity of video content captured by the node, and the VSN topology. In our future work, in addition to the encoding parameters, we take into account the spatial and temporal complexity of the content. We also plan to include more complex VSN topology in our study, where some nodes may need to send their data through intermediate nodes. Thus, in order to find the optimal configuration for each node, the nodes' reception power consumption needs to be taken into account. Also, in order to comply with the bandwidth constraint, we may need to consider using different QP settings for different VSN nodes.

Disclaimer

The statements made herein are solely the responsibility of the authors.

Conflict of Interests

The authors declare that they have no conflict of interests.

Acknowledgment

This work was supported by NPRP Grant no. NPRP 4-463-2-172 from the Qatar National Research Fund (a member of the Qatar Foundation).

References

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.
- [2] N. Xu, S. Rangwala, K. K. Chintalapudi et al., "A wireless sensor network for structural monitoring," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pp. 13–24, New York, NY, USA, November 2004.
- [3] H.-R. Lee, K.-Y. Chung, and K.-S. Jhang, "A study of wireless sensor network routing protocols for maintenance access hatch condition surveillance," *Journal of Information Processing Systems*, vol. 9, no. 2, pp. 237–246, 2013.
- [4] H. Ahn, H. Kim, and J. R. Park, "Smart monitoring of indoor asbestos based on the distinct optical properties of asbestos from particulate matters," *Journal of Convergence*, vol. 5, no. 3, pp. 22–25, 2014.
- [5] V. Shnayder, B.-R. Chen, K. Lorincz, T. R. F. F. Jones, and M. Welsh, "Sensor networks for medical care," in *Proceedings of the 3rd International Conference Embedded Networked Sensor Systems (SenSys '05)*, p. 314, New York, NY, USA, November 2005.
- [6] T. Naumowicz, R. Freeman, H. Kirk et al., "Wireless sensor network for habitat monitoring on Skomer Island," in *Proceedings of the 35th Annual IEEE Conference on Local Computer Networks (LCN '10)*, pp. 882–889, IEEE, Denver, Colo, USA, October 2010.
- [7] H. Ahn, M. Ju, D. Yoo, H. Kim, and Y. Kim, "Data analysis of fish species change depending on existence of wetland at Lake Paro Upstream for the wireless monitoring of ecosystem," *The Journal of Convergence*, vol. 5, no. 4, pp. 23–27, 2014.
- [8] J. C. Augusto, V. Callaghan, D. Cook, A. Kameas, and I. Satoh, "Intelligent environments: a manifesto," *Human-Centric Computing and Information Sciences*, vol. 3, no. 1, article 12, 2013.
- [9] E.-J. Lee, C.-H. Kim, and I. Y. Jung, "An intelligent green service in internet of things," *The Journal of Convergence*, vol. 5, no. 3, pp. 4–8, 2014.
- [10] S. C. Mukhopadhyay, A. Gaddam, and G. S. Gupta, "Wireless sensors for home monitoring—a review," *Recent Patents on Electrical Engineering*, vol. 1, no. 1, pp. 32–39, 2010.
- [11] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks*, vol. 51, no. 4, pp. 921–960, 2007.
- [12] S. Soro and W. Heinzelman, "A survey of visual sensor networks," *Advances in multimedia*, vol. 2009, Article ID 640386, 21 pages, 2009.
- [13] A. Seema and M. Reisslein, "Towards efficient wireless video sensor networks: a survey of existing node architectures and proposal for a flexi-WVSNP design," *IEEE Communications Surveys and Tutorials*, vol. 13, no. 3, pp. 462–486, 2011.
- [14] W.-C. Feng, E. Kaiser, W. C. Feng, and M. L. Baillif, "Panoptes: scalable low-power video sensor networking technologies," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 1, no. 2, pp. 151–167, 2005.
- [15] P.-Y. Chen, W.-S. Lee, and C.-F. Huang, "Design and implementation of a real time video surveillance system with wireless

- sensor networks,” in *Proceedings of the Vehicular Technology Conference (VTC Spring '08)*, pp. 218–222, Singapore, May 2008.
- [16] Z. Shuai, S. Oh, and M.-H. Yang, “Traffic modeling and prediction using camera sensor networks,” in *Proceedings of the 4th ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC '10)*, pp. 49–56, September 2010.
- [17] X. Ren and Z. Yang, “Research on the key issue in video sensor network,” in *Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT '10)*, pp. 423–426, IEEE, Chengdu, China, July 2010.
- [18] N. Tezcan and W. Wang, “Self-orienting wireless multimedia sensor networks for maximizing multimedia coverage,” in *Proceedings of the IEEE International Conference on Communications (ICC '08)*, pp. 2206–2210, Beijing, China, May 2008.
- [19] S. M. A. Zaidi, J. Jung, and B. Song, “Prioritized multipath video forwarding in WSN,” *Journal of Information Processing Systems*, vol. 10, no. 2, pp. 176–192, 2014.
- [20] C. B. Margi, R. Manduchi, and K. Obraczka, “Energy consumption tradeoffs in visual sensor networks,” in *Proceedings of the 24th Brazilian Symposium on Computer Networks (SBRC '06)*, Curitiba, Brazil, June 2006.
- [21] J. J. Ahmad, H. A. Khan, and S. A. Khayam, “Energy efficient video compression for wireless sensor networks,” in *Proceedings of the 43rd Annual Conference on Information Sciences and Systems (CISS '09)*, pp. 629–634, Baltimore, Md, USA, March 2009.
- [22] B. A. B. Sarif, M. T. Pourazad, P. Nasiopoulos, and V. C. M. Leung, “Encoding and communication energy consumption trade-off in H.264/AVC based video sensor network,” in *Proceedings of the IEEE 14th International Symposium and Workshops on a World of Wireless, Mobile and Multimedia Networks (WoWMoM '13)*, pp. 1–6, IEEE, Madrid, Spain, June 2013.
- [23] J. -H. Chang and L. Tassiulas, “Maximum lifetime routing in wireless sensor networks,” *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 609–619, 2004.
- [24] A. Sinha and D. K. Lobiyal, “Performance evaluation of data aggregation for cluster-based wireless sensor network,” *Human-Centric Computing and Information Sciences*, vol. 3, article 13, 2013.
- [25] S.-K. Bae, “Power consumption analysis of prominent time synchronization protocols for wireless sensor networks,” *Journal of Information Processing Systems*, vol. 10, no. 2, pp. 300–313, 2014.
- [26] I. E. Richardson, *The H.264 Advanced Video Compression Standard*, John Wiley & Sons, 2nd edition, 2010.
- [27] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.
- [28] B. A. B. Sarif, M. T. Pourazad, P. Nasiopoulos, and V. C. M. Leung, “Analysis of power consumption of H.264/AVC-based video sensor networks through modeling the encoding complexity and bitrate,” in *Proceedings of the 18th International Conference on Digital Society (ICDS '14)*, Barcelona, Spain, 2014.
- [29] P. M. Kuhn, “A complexity analysis tool: iprof (version 0.41),” Tech. Rep. ISO/IEC JTC1/SC29/WG11/M3551, 1998.
- [30] B. A. B. Sarif, M. T. Pourazad, P. Nasiopoulos, and V. C. M. Leung, “A new scheme for estimating H.264/AVC-based video sensor network power consumption,” in *Proceedings of the 2015 World Congress on Information Technology Applications and Services*, Jeju, Republic of Korea, 2015.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

