DEVS-Based Evaluation for a Proper Selection of Compression and Encryption in Ubiquitous Computing Environment

Ki-Woong Park and Kyu-Ho Park
Computer Engineering Research Laboratory
Department of Electrical Engineering and Computer Science
Korea Advanced Institute of Science and Technology
{woongbak,kpark}@core.kaist.ac.kr

Abstract

In an attempt to achieve an efficient and secure communication in ubiquitous computing environment, we found that a proper selection of compression and encryption algorithms is a challenging issue. Compression can be beneficial to reduce the number of bits transmitted. If the energy required to compress data is less than the energy required to send it, there is a net energy savings and an increase in battery life for portable computers. On the other hand, encryption is essential operations to provide a secure communication. However, the mobile devices, usually with very limited resources and battery power, are subject to the problem of energy consumption due to compression and encryption algorithms. They consume a significant amount of computing resources such as CPU time, memory, and battery power. In this paper, we present the results of DEVS-based evaluation with symmetric key algorithms (AES, RC4) and compression algorithms (Zlib, LZO, Bzip) that are commonly suggested in terms of processing time and consumed energy.

1 Introduction

Ubiquitous computing environment are usually built with a large number of inexpensive, small, and battery-powered devices. They have been used for a wide variety of applications such as environment monitoring, health monitoring, military sensing and tracking, etc [1]. In hostile environments such as battlefield surveillance, an adversary can eavesdrop on traffic, inject new messages, and replay old messages. Therefore, it is necessary to incorporate appropriate secure mechanisms into ubiquitous computing environment. However, given the stringent constraints on processing power, memory, bandwidth, and energy consumption, it is very difficult to design suitable secure mechanisms for ubiquitous sensor networks. This paper aims at the evaluation of the processing time and consumed energy for compression (LZO, Bzip, and Zlib) and encryption (AES, RC4, RSA) algorithms for a proper selection of compression and encryption in ubiquitous computing environment.

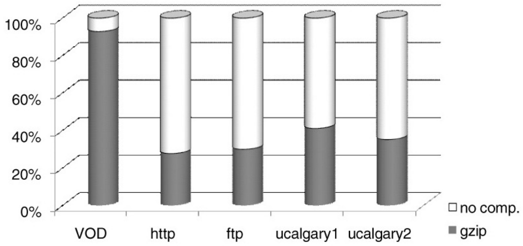


Figure 1. Compression rate evaluation of internet services using Ethereal toolkit.

Implementations which made concessions in compression ratio to improve performance might be modified to provide an overall energy savings. For example, PANDA [4] developed by KAIST CORE Lab. consist of an 8 MHz 8-bit Atmel ATMEGA128L CPU with only 4 Kbyte of RAM space for data, 128 Kbyte of program memory, and 512 Kbyte flash memory. This leaves very limited resources for the necessary security components in the mobile devices. The constraints posed by the mobile devices hamper to deploy most of the traditional security primitives and protocols. We took an experiment about the entropy of Internet traffic using Ethereal S/W to evaluate the compression ratio. Fig. 1 illustrates the compression ratio of Internet traffic using Zlib compression algorithm. The traffic of VOD services which include video and audio data has a low compression ratio (less than 10%). On the other hand, http and

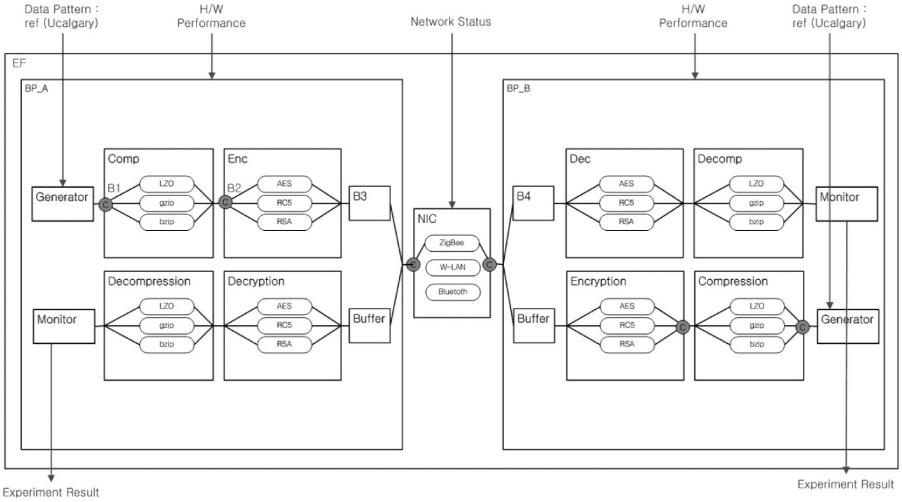


Figure 2. Overall DEVS-Based Evaluation Framework

ftp services show much higher compression ratio (higher than 50%) in comparison to the VOD services. The reason of the fact is that most of the video and audio CODECs are designed considering the network efficiency and compression so that it can provide efficient communication mechanism. From this experiment, we could convince the fact that the compression algorithms are beneficial to perform additional computation to reduce the number of bits transmitted. Therefore, a proper selection of compression and encryption algorithms is a challenging issue. In this paper, we present the results of DEVS-based evaluation with symmetric key algorithms (AES, RC4) and compression algorithms (Zlib, LZO, Bzip) that are commonly suggested in terms of processing time and consumed energy. The remainder of the paper is organized as follows: In Section 2, we present our DEVS-Based evaluation framework for a proper selection of compression and encryption algorithms. In Section 3, we present a design of an experimental framework including input data presentation and performance index. In Section 4, we evaluate the performance of several compression (LZO, Zlib, Bzip) and encryption (AES, RC4, RSA) algorithms. Finally, in Section 5, we present our conclusions.

2 DEVS-Based Evaluation Framework

The goal of this paper is to present the results of DEVS-based evaluation with encryption algorithms (AES, RC4, RSA) and compression algorithms (Zlib, LZO, Bzip) that are commonly suggested in terms of processing time and consumed energy. To evaluate them, we constructed the DEVS-Based evaluation framework which models computing power and energy consumption of our mobile platforms (UFC, PANDA).

2.1 Evaluation Framework Architecture

The overall architecture for evaluation of compression and encryption algorithms in UFC and PANDA with multi modal communication module (W-LAN, Bluetooth, and ZigBee) is presented in Fig 2. The overall experiment framework consists of an experiment framework (EF), sender (BP_A), communication (Comm), receiver (BP_B) modules. The functionality of the modules is described as follows;

- **EF Module:** It is an experiment framework which consists of two modules. They are a generator module and a monitor module. The first one generates network traffic considering data characteristic and frequency based on Ucalgary [9]. The generated data are transferred to BP_A module. The second module has a function to monitor of power consumption of each modules and summarize the evaluation results.
- **BP_A Module:** It receives the generated data from EF module and compresses/encrypts the received data in the compression (Comp) module and encryption (Enc) module respectively. The function of the modules is described as follows;
    - B1 Module: receives the generated data from EF modules and transmits to the compression (Comp) module.

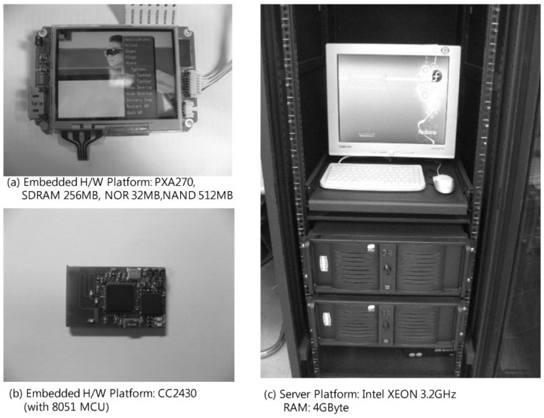
- Comp Module: compresses the received data from B1 module. Applied compression algorithms (LZO, Zlib, Bzip) are selected by the experiment parameter.
- B2 Module: receives the compressed data from the compression (Comp) modules and transmits to the encryption (Enc) module.
- Enc Module: encrypts the received data from B2 module. Applied encryption algorithms (AES, RSA, RC4) are selected by the experiment parameter.
- B3 Module: receives the encrypted data from the encryption (Enc) modules and transmits to the communication (Comm) module.

- Comm: It is a communication modeling module. The total delivery time and latency of communication module is modeled by communication experiment using H/W platform developed by KAIST CORE Lab.
- **BP_B Module:** It receives the compressed and encrypted data from BP_A module and decompresses/decrypts the received data in the decompression (Decomp) module and decryption (Dec) module respectively. The function of the modules is described as follows;
  - B4 Module: receives the compressed and encrypted data from the communication (Comm) module and transmits to the decryption (Dec) module.
  - Dec Module: decrypts the received data from B4 module. Applied decryption algorithms (AES, RC4, RSA) are selected by the experiment parameter.
  - B5 Module: receives the decrypted data from the decryption (Dec) modules and transmits to the decompression (Decomp) module.
  - Decomp Module: decompresses the received data from B5 module. Applied decompression algorithms (LZO, Zlib, Bzip) are selected by the experiment parameter.

2.2 Modeling with real equipments

To implement our evaluation framework with our designed modeling and simulation, we take a step as follows,

- Modeling of H/W platform characteristics: To achieve it, we evaluate the processing time and consumed energy for compression/encryption algorithms for several H/W platforms. Fig. 3 shows our experiment equipment to evaluate the processing time and consumed energy in several platforms.


(a) Embedded H/W Platform: PXA270, SDRAM 256MB, NOR 32MB,NAND 512MB

(b) Embedded H/W Platform: CC2430 (with 8051 MCU)

(c) Server Platform: Intel XEON 3.2GHz RAM: 4GByte

Figure 3. Experiment equipment to evaluate the processing time and consumed energy in several platforms

- Evaluation of the performance characteristics of compression and encryption algorithms: To evaluate the performance characteristics of compression and encryption algorithms, we constructed a pseudo H/W platform, which are coupled to a load generator. The objective of the pseudo devices is to simulate the processing and communications resources anticipated in a full implementation.

Fig. 4 shows the overall experimental environment. The pseudo devices have operation times that are similar to actual operation times (for example, those that result from communications, encryption and decryption, and compression processing).The pseudo devices are modeled on the cryptography operation (AES, RC4, RSA) times, the data rate, the detection ratio, and the delivery latency of ZigBee, W-LAN, and W-LAN communications. Moreover, these pseudo devices are connected to our simulation framework and receive control signals from the load generator. The load generator is a module that generates control signals to produce compression and encryption messages by using a random generator that models the communication pattern of the mobile devices [7].

3 Experiment results and analysis with real equipments

To generate an input data set for experiment, we acquired the files that we studied from well-known compression corpora, including the Canterbury Corpus [8] and Calgary Corpus [9]. The Canterbury Corpus is a new corpus introduced

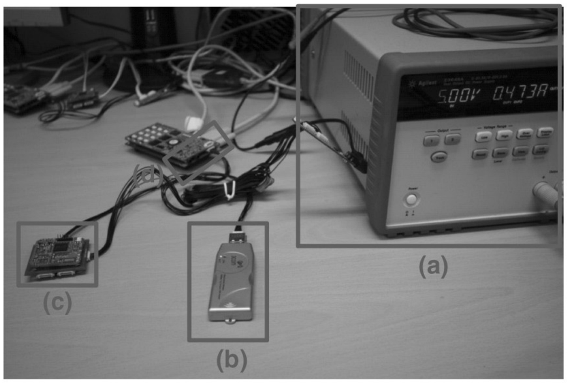


Figure 4. (a) Power supply & Monitor equipment to evaluate the consumed energy (b) W-LAN Device (c) Bluetooth Module (d) ZigBee Module

to replace the old Calgary Corpus. The Calgary Corpus, despite its age, remains a well-respected corpus that is frequently used for the comparison of compression algorithms. Both corpora include both English text (bibliography, book, paper, etc.) and non-text sources (picture, object code, geophysical data, etc.) as follows;

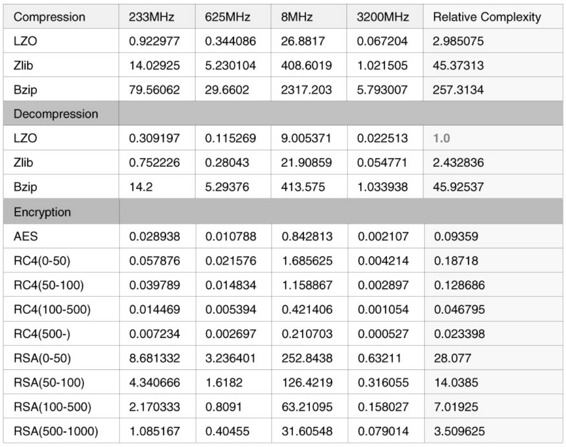
- 1MB English text from "Calgary Corpus" - A novel and structured bibliography
- 1MB web data from most popular sites according to "Lycos Top 50" searches

Using H/W platform described in the previous section, we measures the processing time of compression/encryption algorithm on the embedded platforms and the server platform. Fig. 5 compares the processing time of LZO, Zlib, and Bzip as a compression algorithm, AES, RC4, and RSA as an encryption algorithm. Each result present the processing time (unit micro second) to process one bit, relative complexity describes the processing complexity assumed that LZO algorithm has one complexity.

Fig. 6 compares the consumed energy of LZO, Zlib, and Bzip as a compression algorithm, AES, RC4, and RSA as an encryption algorithm. Each result present the consumed energy (Unit: micro J) to process one bit, relative complexity stands for the processing complexity assumed that LZO algorithm has one complexity.

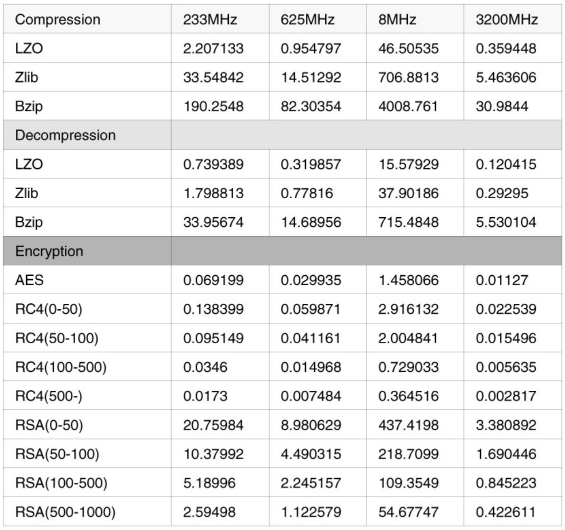
4 DEVS-Based evaluation and analysis

The performance index to analysis the processing power and consumed energy of compression/encryption algorithm



Compression	233MHz	625MHz	8MHz	3200MHz	Relative Complexity
LZO	0.922977	0.344086	26.8817	0.067204	2.985075
Zlib	14.02925	5.230104	408.6019	1.021505	45.37313
Bzip	79.56062	29.6602	2317.203	5.793007	257.3134
Decompression					
LZO	0.309197	0.115269	9.005371	0.022513	1.0
Zlib	0.752226	0.28043	21.90859	0.054771	2.432836
Bzip	14.2	5.29376	413.575	1.033938	45.92537
Encryption					
AES	0.028938	0.010788	0.842813	0.002107	0.09359
RC4(0-50)	0.057876	0.021576	1.685625	0.004214	0.18718
RC4(50-100)	0.039789	0.014834	1.158867	0.002897	0.128686
RC4(100-500)	0.014469	0.005394	0.421406	0.001054	0.046795
RC4(500-)	0.007234	0.002697	0.210703	0.000527	0.023398
RSA(0-50)	8.681332	3.236401	252.8438	0.63211	28.077
RSA(50-100)	4.340666	1.6182	126.4219	0.316055	14.0385
RSA(100-500)	2.170333	0.8091	63.21095	0.158027	7.01925
RSA(500-1000)	1.085167	0.40455	31.60548	0.079014	3.509625

Figure 5. Processing times of compression/encryption for each algorithm and operation environment



Compression	233MHz	625MHz	8MHz	3200MHz
LZO	2.207133	0.954797	46.50535	0.359448
Zlib	33.54842	14.51292	706.8813	5.463606
Bzip	190.2548	82.30354	4008.761	30.9844
Decompression				
LZO	0.739389	0.319857	15.57929	0.120415
Zlib	1.798813	0.77816	37.90186	0.29295
Bzip	33.95674	14.68956	715.4848	5.530104
Encryption				
AES	0.069199	0.029935	1.458066	0.01127
RC4(0-50)	0.138399	0.059871	2.916132	0.022539
RC4(50-100)	0.095149	0.041161	2.004841	0.015496
RC4(100-500)	0.0346	0.014968	0.729033	0.005635
RC4(500-)	0.0173	0.007484	0.364516	0.002817
RSA(0-50)	20.75984	8.980629	437.4198	3.380892
RSA(50-100)	10.37992	4.490315	218.7099	1.690446
RSA(100-500)	5.18996	2.245157	109.3549	0.845223
RSA(500-1000)	2.59498	1.122579	54.67747	0.422611

Figure 6. Consumed energy of compression/encryption for each algorithm and operation environment

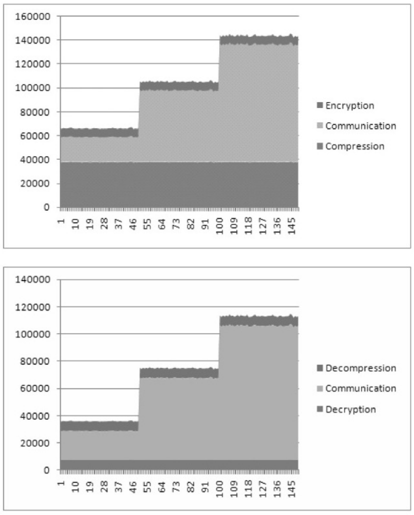


Figure 7. The variation of the energy consumption with varying the communication modules, top: sender, bottom: receiver

consists of three one as follows;

- Total Delivery Time
- Required Energy

4.1 Energy aspect

Fig. 7 shows the variation of the energy consumption with varying the communication modules (W-LAN ⟶ Bluetooth ⟶ ZigBee). ZigBee has the lowest energy efficiency, and W-LAN has the highest energy efficiency. The reason of the fact is that the required energy transmitting one bit over communication module, W-LAN has the highest energy efficiency comparing the ZigBee or Bluetooth modules.

Fig. 8 shows the required energy consumption with varying the compression algorithms (LZO ⟶ Zlib ⟶ Bzip). Even though LZO has the highest computational efficiency, the compression ratio of LZO is lower than other compression ratio considering the addition of the network energy of LZO. Therefore, the computing power of each entity and communication status should be considered to apply the suitable compression and encryption algorithms.

Fig. 9 shows the required energy consumption with varying the encryption algorithms (AES ⟶ RC4 ⟶ RSA). In a consumed energy of the encryption algorithms, our experiments show that the RC4 is fast and energy efficient for compressing large packets comparing AES and RSA. However, AES was more efficient than RC4 for a communication module providing a lower bandwidth (ZigBee) than W-LAN and Bluetooth. Especially, the consumed energy of RSA has 250 times consumed energy so that it is inapplicable to a resource constrained device such as USN.

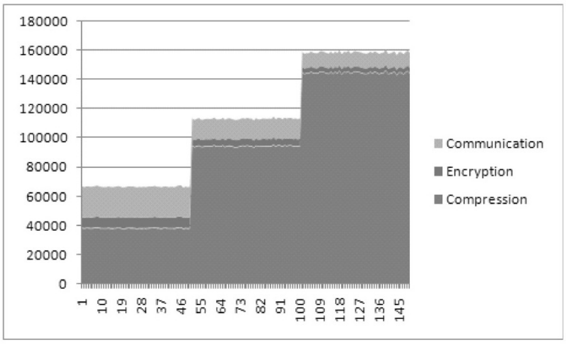


Figure 8. The variation of the energy consumption with varying the compression modules

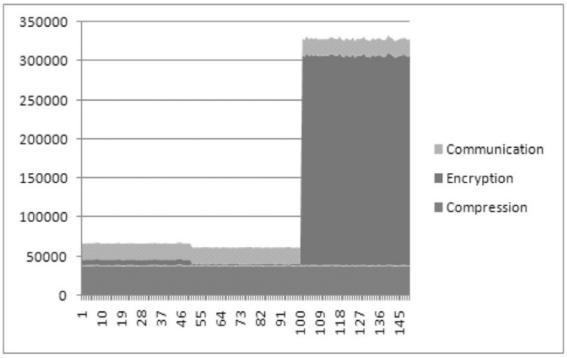


Figure 9. The variation of the energy consumption with varying the encryption modules

4.2 Processing time aspect

Fig. 10 shows the variation of the processing time with varying the communication modules (W-LAN ⟶ Bluetooth ⟶ ZigBee). ZigBee has the longest delivery time sue to the physical limitation of ZigBee, and W-LAN has the shortest delivery time.

Fig. 11 shows the required processing time with varying the compression algorithms (LZO ⟶ Zlib ⟶ Bzip).

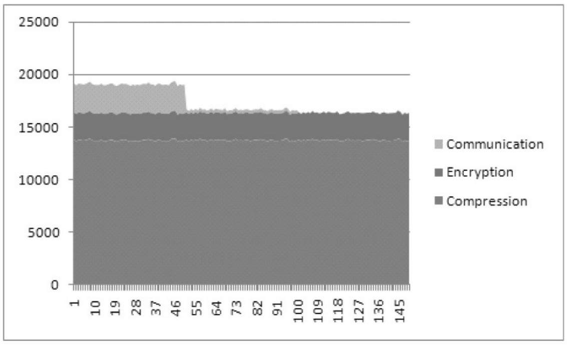


Figure 10. The variation of the total delivery time with varying the communication modules

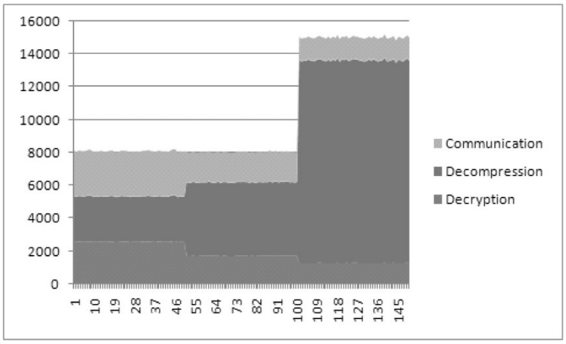


Figure 11. The variation of the processing time with varying the communication modules

Even though LZO has the highest computational efficiency, the compression ratio of LZO is lower than other compression ratio considering the addition of the total delivery time of the communication module. Therefore, the computing power of each entity and communication status should be considered to apply the suitable compression and encryption algorithms.

Fig. 12 shows the required processing time with varying the encryption algorithms (AES ⟶ RC4 ⟶ RSA). In a processing time of the encryption algorithms, our experiments show that the RC4 is fast and energy efficient for compressing large packets comparing AES and RSA. However, AES was more efficient than RC4 for a shorter message less than 4KB. In case of RSA, The asymmetric operation characteristics make encryption processing time using the private key much longer than the decryption operation time.

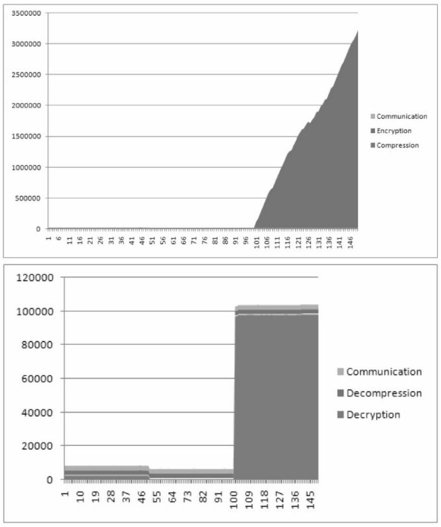


Figure 12. The variation of the processing time with varying the encryption modules, top: sender, bottom: receiver

4.3 Analysis of experiment result in heterogeneous h/w platform environment

Fig. 13 shows the comparison of the consumed energy and processing time for the best case and the worst case (suitable compression algorithm, encryption algorithm). In case of the ideal case (red stick) can improve the energy consumption till 85%. On the other hand, the worst case which means that an incorrect compression/encryption algorithm is applied to the system makes it possible to reduce energy efficiency or increase the total delivery time.

5 Conclusion

In this paper, we present a DEVS-Based evaluation in order to provide a proper selection of symmetric/asymmetric key algorithms and compression algorithms. we presented our evaluation of the processing time and consumed energy for compression (LZO, Zlib, Bzip) and encryption (AES, RC4, RSA) algorithms. The performance metrics were encryption/compression processing time, energy cost. In a processing time of the encryption algorithms, our experiments show that the RC4 is fast and energy efficient for encrypting large packets comparing AES and RSA. However, AES was more efficient than RC4 for a communica-

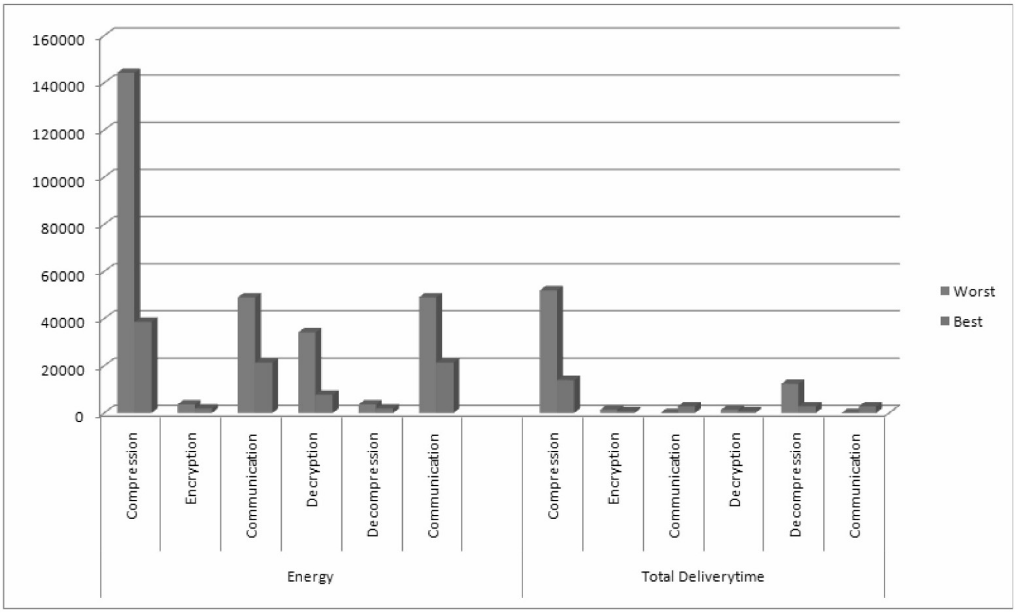


Figure 13. The comparison of the consumed energy and processing time for the best case and the worst case (suitable compression algorithm, encryption algorithm)

tion module providing a lower bandwidth (ZigBee) than W-LAN and Bluetooth. From our results, it appears that we can save energy by using a combination of compression and encryption algorithms considering packet size and processing power of each network entities (Sender and receiver). The tradeoffs between the processing time and the energy consumption are not completely clear. As our future work, we are studying the relationship between compression algorithms and encryption algorithms.

References

[1] Bo Sun, Chung-Chih Li, Kui Wu, Yang Xiao, A lightweight secure protocol for wireless sensor networks, Computer Communications 29 (2006) 2556-2568

[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Computer Networks 38 (4) (2002) 393-422.

[3] KENNETH C. BARR and KRSTE ASANOVI C, Energy-Aware Lossless Data Compression, ACM Transactions on Computer Systems, Vol. 24, No. 3, August 2006, Pages 250-291

[4] Ki-Woong Park, S.S. Lim, H.C. Seok, and K.H. Park, ”Ultra-Low-Power Security Card, PANDA, for PKI-based Authentication and Ubiquitous Services,” Proceedings of Conference on Next Generation Computing, November 2006, pp.367-373

[5] Chandra Krintz and Sezgin Sucu, ”Adaptive On-the-Fly Compression,” IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, JANUARY 2006

[6] R. CHANDRAMOULI, S. BAPATLA, and K. P. SUBBALAKSHMI, ”Battery Power-Aware Encryption,” ACM Transactions on Information and System Security, Vol. 9, No. 2, May 2006

[7] Prasithsangaree and P. Krishnamurthy, ”Analysis of Energy Consumption of RC4 and AES Algorithms in Wireless LANs,” IEEE International Conference on GLOBECOM 2003

[8] A.Gellert, and L.Vintan, ”Person Movement Prediction Using Hidden Markov Models,” Proc. of Studies in Informatics and Control, Vol. 15, No. 1, ISSN 1220-1766 (ISI Thomson INSPEC), National Institute for Research and Development in Informatics, Bucharest, 2006.

[9] Canterbury corpus, http://corpus.canterbury.ac.nz/+, 2005.

[10] Calgary corpus, http://links.uwaterloo.ca/calgary.html+,2005.