This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2010 proceedings

Framework for Energy-aware Lossless Compression in Mobile Services: the Case of E-mail

Yu Xiao, Matti Siekkinen, and Antti Ylä-Jääski
Department of Computer Science and Engineering
Aalto University
Espoo, Finland
{yu.xiao, matti.siekkinen, antti.yla-jaaski}@tkk.fi

***Abstract*—Energy consumption caused by wireless transmission poses a big challenge to the battery lifetime of mobile devices. While the potential of using lossless compression for saving energy has been long acknowledged, no general solution has been proposed for applying lossless compression to energy adaptation for mobile services. We propose a proxy-based energy adaptation framework, in which the data to be transmitted is losslessly compressed on a proxy server according to context-aware policies. The context includes factors relevant to computational and communication cost, as well as the user's preferences. We showcase a context-aware policy which aims at minimizing client-side energy consumption caused by transmission and decompression. Using our framework, we implement an energy-aware mobile e-mail service, and present power measurement results that show significant energy savings.**

Keywords- Energy-aware; Lossless compression;Mobile Service

I. INTRODUCTION

As mobile services that use wireless data transmission have gained popularity, the energy consumption caused by wireless data transmission has become a real challenge to the battery lifetime of mobile devices. Energy consumption of wireless data transmission highly depends on the amount of data to be transmitted. It is also known that wireless transmission of a single bit can consume over 1000 times more energy than a single 32-bit computation [1]. Therefore using compression to reduce the amount of transmitted data is desirable from the perspective of energy consumption.

The efficiency of energy saving using compression depends on the tradeoff between the computational overhead caused by the compression/decompression operations and the energy saved in network transmission. The computational cost relates to the computational complexity of compression algorithms and the processing performance of the mobile devices. The energy reduction in transmission relies on the reduced traffic size and the network conditions during transmission. The reduction in data size is determined by the compression ratio which varies with the compression algorithm and data type, while the network conditions can be evaluated by the network throughput, signal-to-noise-ratio (SNR), or bit error rate. The network transmission consumes relatively less energy under better network conditions such as higher network throughput [9] and higher SNR [2]. The estimation of network conditions is therefore essential for the estimation of communication cost and, further, the efficiency of energy adaptation.

Many solutions have been published since the first proposal [1] of using compression for energy savings. Each addresses an individual part of the whole problem, and there is no general solution about how to apply lossless compression to energy-aware mobile services. For example, [1] analyzed the communication-to-computation energy ratio for different compression algorithms, but only provided quantitative results of communication cost at 2.8Mb/s and 5.6Mb/s and did not propose any model for estimating communication cost under different network data rates.

In this paper, we propose a proxy-based framework for mobile services using energy-aware lossless compression, and exemplify the approach through a proof-of-concept implementation of an energy-aware mobile e-mail service. The framework includes two parts, a proxy responsible for the decision-making of compression-based energy adaptations and the execution of compression, and mobile clients in charge of decompression. We model the decision-making based on the tradeoff between computational and communication cost. To the best of our knowledge, ours is the first approach towards a complete framework that considers all the relevant factors of computational and communication cost such as data characteristics, compression algorithm characteristics, and network conditions.

Our contribution includes, but is not limited to, the following three aspects.

- Proposing a practical solution adopting a decision-making mechanism based on context-aware policies. The decision-making mechanism supports the selection among multiple compression algorithms which is based on the estimation of computational and communications cost.
- Taking into account the compression ratios of different data types, and the client context such as the energy utility of decompression when estimating the computational cost.
- Taking network conditions in terms of network throughput and network signal-to-noise-ratio (SNR) into account when estimating the communication cost.

The remainder of this paper is structured as follows. We review the related work in Section 2. Section 3 illustrates the system model of our energy-aware lossless compression framework, and explains the selection of data compression mechanisms. The implementation and evaluation of our

This work was supported by TEKES as part of the Future Internet program of TIVIT (Finnish Strategic Centre for Science, Technology and Innovation in the field of ICT).

978-1-4244-6404-3/10/$26.00 ©2010 IEEE

framework is described in Section 4, taking a mobile e-mail service as an example. Finally, we conclude the paper and present our future work in Section 5.

II. Related Work

Data compression has been applied for saving storage space [3], shortening network latency [4, 5], saving computing power [6, 7], and reducing transmission cost [2, 8] in mobile computing. The core issues of these applications consist of the performance of different compression algorithms, and the compression mechanisms adopted by different application scenarios. The performance of compression algorithms is evaluated in terms of compression ratio, compression time, decompression time, memory allocation, and the energy consumption of compression and decompression, while the compression mechanisms determine when and where to execute compression, which compression algorithm to use, and who makes these decisions.

The comparison of the compression performance of a number of popular compression algorithms such as *compress*, *gzip*, *bzip2*, *LZO*, and *zlib* has been presented in [1, 5, 8]. According to the results, there is no single compression algorithm that performs the best on all types of data. This means that in order to gain a high compression performance, the compression algorithm should be chosen according to data type. In addition, the priority of different aspects of compression performance varies from application scenario to another. For instance, Krintz and Sucu [5] aimed at improving network transmission performance with the on-the-fly compression of network traffic streams, and thus chose the compression algorithms to be used based on the compression ratio, the compression time and the decompression time. In this paper, we focus on energy savings for data receivers. Hence, we select the compression algorithms based on compression ratio and the decompression energy consumption.

Compression mechanisms can be divided into always-compression and adaptive compression. The latter one only compresses data when certain requirements are met. Lossless data compression does not always result in energy savings to the data receivers in a network transmission due to the computational cost of decompression. If aiming for energy savings, always-compression is better to be replaced with an adaptive compression scheme based on the tradeoff between the computational cost and the communication cost saved by the reduced data size does.

The performance of decompression operations on data receivers depends on the processing performance of the receivers and the computational complexity of the compression algorithms. Xu et al. [8] assumed that the rate of energy consumption is fixed during decompression, and they calculated the energy consumption of decompression as the product of assumed fixed rate of energy consumption and the predicted decompression time. As decompression time depends on the CPU load [5], which is difficult to predict beforehand, we propose predicting the energy consumption of decompression based on the compressed file size and the energy utility of decompression, instead of decompression time.

The communication cost of receiving compressed data relies on multiple underlying factors such as network signal strength [2] and network throughput. However, most of the previous work only takes some of these network conditions into account and ignores the others. For example, [8] assumed that the energy consumption of downloading a unit size of data is fixed in spite of network conditions. Maddah and Sharafeddine [2] compared the energy savings of using compression on different network signal levels, but did not take other network conditions such as network throughput into account. Our work overcomes the limitations of previous work and takes network throughput and possible network fluctuation into account when predicting communication cost.

Previous work [2, 8] usually adopted threshold-based adaptive policies for energy-aware data compression. For instance, [2] chose to compress data when the network signal level was lower than a certain threshold, while [8] did a block-by-block compression when the block size was bigger than a certain threshold. Differently from the adaptations based on a single threshold, of the decision we make on when to compress and what compression algorithm to use depends on the analysis of compression effectiveness defined in Section 3. The compression effectiveness reflects the tradeoff between computational and communication cost.

When designing system architecture for energy-aware data compression, it is important to define where to execute compression and who makes the decisions on compression. For example, in the proxy-based architecture used in [6, 7], the proxies transcoded the video streams into lower playback quality levels, and chose the targeted quality level based on the clients' battery budget and playback power consumption. Similarly, we adopt a proxy-based structure. However, differently from [6, 7], we take communication cost in different network contexts into account when deciding whether to compress. In addition, we choose the most well-suited compression algorithm for the data at hand and take the availability of compression algorithms on mobile clients into consideration.

III. Mobile Services Using Energy-aware Lossless Compression

The aim of our framework is to reduce the energy consumption of mobile devices by compressing the data prior to the data transmission to/from the mobile devices. In this paper, we focus on the scenarios in which mobile devices receive data transmitted from the network. Accordingly, we propose a proxy server performing lossless compression before forwarding the data to the mobile clients. The proxy server is also responsible for the selection of the compression mechanism. The selection will take into account both the transmission and decompression energy usage on the client side. In this section, we will first present the system model of our framework, and then explain how to select the compression mechanism automatically and transparently.

A. System Model

We assume a system of servers such as e-mail and media streaming servers, and mobile clients that request content from the servers through intermediate proxy servers.

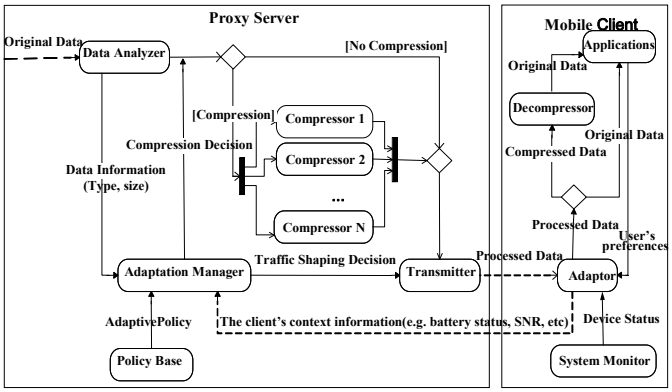

Figure 1. Framework of energy-aware lossless compression in mobile services.

The mobile clients are powered by batteries, while the proxy servers use external power supplies. The status of hardware, operating system, wireless networks, and applications running on the mobile device are defined as the context information of a mobile client. Examples of this context information are the residual battery status, the list of compression algorithms supported by the hardware/OS, and the network signal-to-noise-ratio. The user's preferences included in the status of applications describe the compression mechanisms that are preferred by the user. For instance, a user may prefer compressing all the data only when the battery status is low.

The system model of our framework as illustrated in Fig.1 includes two parts, namely, the proxy server and the mobile client. The mobile client provides its context information to the proxy server upon request, while the proxy server makes compression decisions based on the client's context information. The proxy server can be further abstracted into four components, namely, data analyzer, policy base, adaptation manager, and network transmitter. The data analyzer detects the data type and data size. The policy base is a container of adaptive policies which describe what compression decisions to use under what conditions. Based on the predefined adaptive policies stored in the policy base, the adaptation manager makes compression decisions using the data type, the data size and the context information of the mobile client as its input.

There are two system components that collect the context information from the mobile device, namely, the system monitor and adaptor, residing on the mobile client. The system monitor collects and forwards the device status to the adaptor, and the adaptor generates messages by combining the device status with the user's preferences obtained from applications. In addition, the adaptor is also responsible for sending the messages to the proxy server, and invoking the corresponding decompressor to restore the received data.

On the proxy server, the compression decisions made by the adaptation manager describe the compression algorithm to be used and the traffic shaping instructions which control when and how to transfer data. If no compression is needed, the original data will be forwarded to the transmitter directly. Otherwise, the original data will be first compressed by the specific compressor identified in the compression decision, and then forwarded to the transmitter. The transmitter sends the processed data to the mobile client according to the traffic shaping instructions. For example, it might shape the traffic into bursts [12], or choose to transfer data only when the network signal level is high enough [2].

B. Adaptation Management

As discussed in Section 3.1, the adaptation manager on the proxy server decides whether to compress or not, and which compression algorithm to use. Concerning the energy consumption of mobile devices as data receivers, energy savings can be gained from an adaptation only if the energy consumption of receiving and decompressing the compressed data is less than that of receiving the original data.

Let the original file size be *s*, the file size after compression be s_c, and the compression ratio be *ratio*. According to the definitions in [2, 8], compression ratio can be calculated as the ratio of the compressed file size to the uncompressed file size as shown in (1).

$$\text{ratio} = \frac{\mathrm{s_c}}{\mathrm{s}} \tag{1}$$

In 802.11 WLANs, the energy consumed by receiving a unit of data to the same device varies with network data rates [9]. We hence define the energy utility of transmission on a mobile device as the energy consumed by receiving a unit of data, and denote it by $E_0(r)$ with network data rate *r*. The values of $E_0(r)$ are device-dependent.

We select several network data rates as reference rates, and measure the energy utility of transmission at each reference rate from our experimental devices. During runtime, we predict the network throughput, and compare it with its closest reference rate from the reference rate list. Let the predicted network throughput be r_p, and the closest reference rate be R_r.We use $E_0(R_r)$ as a baseline for estimating the transmission cost at data rate r_p.

Derived from [9] that $E_0(r)$ linearly decreases with *r*, we introduce an adaptive factor α which reflects the linear relationship as shown in (2). The energy consumption of receiving the compressed or the original data to the mobile device can be calculated as (3) and (4) respectively.

$$\alpha = \frac{R_r}{r_p} \tag{2}$$

$$\text{Energy(receiving the compressed data)} = \mathrm{S_c} \times \mathrm{E_0(R_r)} \times \alpha \tag{3}$$

$$\text{Energy(receiving the original data)} = \mathrm{s} \times \mathrm{E_0}(R_r) \times \alpha \tag{4}$$

The energy cost of decompression depends on the computational complexity of the compression algorithm used, and the power characteristics of the hardware device. According to previous work [8], the rate of energy consumption during decompression is stable when the CPU load is stable. Although the decompression time might increase when the CPU load increases, the increase in the decompression time is caused by the sharing of the CPU with

other processes, and the computation of decompression itself can be assumed to be fixed.

Instead of using the uncertain decompression time for estimating decompression energy cost, we define the energy utility of decompression as the energy consumed by restoring a unit of data, and denote it by E_d. According to our empirical power measurement, the energy utility of decompression linearly depends on the compression ratio. For example, we got a linear regression model (5) from the power measurement of decompression using *gunzip* on the Nokia N810. The minimum squared error of (5) is 0.761. The total decompression cost can be expressed as (6).

$$E_d = 0.2635 \times \text{ratio} + 0.4302 \quad (5)$$

$$\text{Energy(decompression)} = E_d \times \text{s} \quad (6)$$

To evaluate the effectiveness of different compression algorithms and to select the algorithms with a maximum energy saving, we define ***Compression Effectiveness*** as the ratio of the energy cost without compression to that with compression. Let the compression effectiveness be *ce*. As shown in (7), the greater *ce* is, the more energy savings will be gained.

$$\text{ce} = \frac{\text{Energy consumption without compression}}{\text{Energy consumption with compression}} = \frac{\text{E}_0(\text{R}_\text{r})\times\alpha\times\text{s}}{\text{s}\times\text{ratio}\times\text{E}_0(\text{R}_\text{r})\times\alpha+\text{E}_\text{d}\times\text{s}} \quad (7)$$

Derived from (7), to achieve energy savings from compression-based adaptation, the decompression-to-transmission energy utility ratio, $\frac{\text{E}_\text{d}}{\text{E}_0(\text{R}_\text{r})}$, must be smaller than a certain threshold. The threshold depends on the compression ratio, and the estimated network throughput as shown in (8).

$$\frac{\text{E}_\text{d}}{\text{E}_0(\text{R}_\text{r})} < \frac{\text{R}_\text{r}}{\text{r}_\text{p}}(1 - \text{ratio}) \quad (8)$$

Based on the above models, we define adaptive policies which guide the decision-making in the adaptation manager during runtime. The adaptive policies define the conditions and the corresponding operations. The conditions are described using context, such as the value of the compression effectiveness, the user's preferences and battery status. Under different conditions, for example, when the value of compression effectiveness is bigger or smaller than 1, the operations are different. The operations could be compression using a certain algorithm or delivering without compression. In Section 4, we will present an example of adaptive policy used in energy-aware mobile e-mail service and a proof-of-concept implementation of it.

IV. ENERGY-AWARE MOBILE E-MAIL

E-mail is one of the most popular Internet services on mobile devices. A mobile e-mail system consists of the Mail User Agent (MUA) running on mobile devices, the Mail Delivery Agent (MDA), and the Mail Transfer Agent (MTA). In modern systems, the MDA can also be replaced by the MTA which is responsible for the delivery of mails. Accordingly, the user creates and sends mails from the sender's MUA, which is also called the e-mail client. After that, the mails will go through the sender's MTA, the recipient's MTA, and finally reach the recipient's MUA.

Currently, various types of files with sizes up to tens of megabytes can be attached to a mail. However, downloading big attachments to mobile devices consumes much energy. Hence, we introduced energy-awareness into a mobile e-mail service, and implemented an energy-aware e-mail MTA and MUA using the framework discussed in Section 3. We will present the implementation in Section 4.1, and the experimental setup and result analysis in Section 4.2.

A. System Implementation

Our energy-aware e-mail service aims at saving energy for the recipient's MUA by delivering compressed attachments to it. The recipient's MTA compresses the attachments before forwarding them to the recipient's MUA according to the predefined adaptive policies. As a proof-of-concept of our framework, the adaptive policies takes into account the data type of attachments, the battery status of the mobile recipient, network conditions, as well as the list of the compression algorithms supported by the mobile recipient.

We implemented the MTA based on Qmail[1], an open source message transfer agent designed for Internet-connected Unix hosts. In addition, we added the functions of context awareness into Claws Mail[2], an open source e-mail client application, and ported it to the Maemo platform.

We adopted POP3 (Post Office Protocol-version 3) for retrieving mails from the MTA in our experiment. The signal sequence of retrieving mails from the MTA is illustrated in Fig. 2. The e-mail recipient's MUA sends a *STAT* message to the MTA asking for the *maildrop* state which means information on new mails. After getting the notification from the MTA, the MUA queries the remaining battery charge of the device and network conditions, and searches for the list of compression algorithms installed in the system. We described the network conditions using the mean and standard deviation of the network signal-to-noise ratio (SNR) in a certain past period. In our experiments, we set it to 30 seconds.

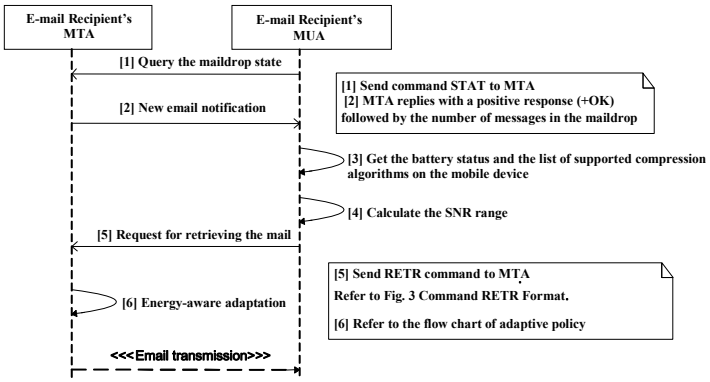

Figure 2. Sequential graph of retrieving emails from Mail Transfer Agent.

[1] Qmail website: www.qmail.org
[2] Claws mail website: www.claws-mail.org

0 1	2 3	4	5	6	7
Battery Status 0..3	SNR Range 0..3	RAR 0..1	Gzip 0..1	Bzip2 0..1	7Zip 0..1

Figure 3. Command RETR Format.

TABLE I. DESCRIPTION OF SNR RANGE

Low	Mean of SNR less than 15
Median	Mean of SNR between 15 and 20
High and Stable	Mean of SNR mean bigger than 20, and the standard deviation of SNR less than 5
High but Fluctuating	Mean of SNR higher than 20, and the standard deviation of SNR bigger than 5

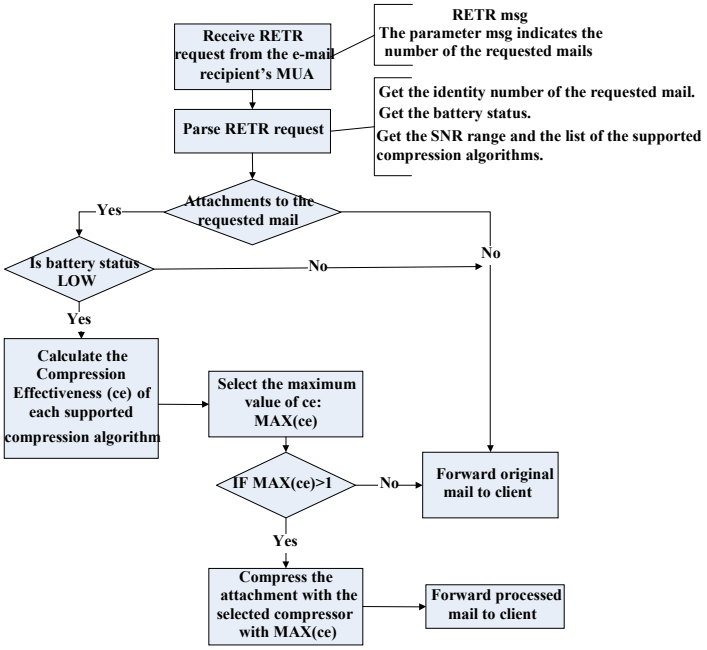

Figure 4. Flow chart of adaptive policy on Mail Transfer Agent (MTA)

All the context information is combined and coded into the *RETR* command as shown in Fig. 3. We used an 8-bit data structure, with the first 2 bits for battery status, the next 2 bits for the SNR range, and the last 4 bits indicating the compression algorithms that are supported. The battery status was divided into 4 levels from 0 to 3, with 0 as the lowest level and 3 as the highest level. We divided the SNR range into four types, namely, Low, Median, High and Stable, and High but Fluctuating. The descriptions of each type are listed in Table I. In addition, if the compression algorithm is supported, the corresponding bit will be set to 1. Otherwise, it is set to 0.

The e-mail recipient's MTA receives the *RETR* command, and parses the context information. Based on the SNR range, the MTA predicted the network throughput of the next transmission based on the history of the network throughput [10] in the same SNR range. We applied in detail the moving average with window size 5. Together with the file information obtained from the file analyzer, the user's preference and the other context information from the MUA, the MTA selects the compression algorithm following the policy described in Fig. 4. In our prototype, we assume that users prefer compression only when the battery status is low, since the decompression requires an additional manual operation.

TABLE II. COMPARISON OF COMPRESSION RATIOS AMONG DIFFERENT FILE EXTENSIONS AND COMPRESSORS

File Extension	*File Size (Byte)*	*RAR*	*Gzip*	*Bzip2*	*7Zip*	*LZO*
.mp3*	6379692	1	0.99	1	1	1
.wav	3089060	0.09	0.13	0.08	0.08	0.22
.jpeg	3446116	1	1	0.99	1	1
.pgm	12582931	0.17	0.33	0.22	0.20	0.44
.bmp*	4149414	0.28	0.30	0.21	0.24	0.55
.xls	9824768	0.21	0.30	0.23	0.15	0.43
.doc*	4168192	0.20	0.24	0.22	0.19	0.28
.txt*	2988578	0.21	0.29	0.19	0.20	0.47
.hlp*	4121418	0.17	0.21	0.17	0.15	0.32
.log*	20617071	0.04	0.07	0.04	0.04	0.12
.exe*	3870784	0.36	0.45	0.44	0.36	0.59
.dll*	3782416	0.49	0.58	0.56	0.48	0.73
.dic*	4067439	0.26	0.26	0.30	0.21	0.43
.pdf*	4526946	0.83	0.85	0.84	0.82	0.88

*Benchmark files from www.maximumcompression.com

TABLE III. ENERGY UTILITY OF NETWORK TRANSMISSION AT DIFFERENT DATA RATES IN WLAN

Reference Data rate (KB/s)	*Energy Utility(mJ/KB)*
16	55.744
32	28.107
64	14.446
96	10.063
128	7.448
160	6.345
192	5.403
224	4.767
256	4.211

We tested the compression ratios using *RAR, Gzip, Bzip2, 7Zip* and *LZO* for selected benchmark files. The compression ratios are stored in the forms of tables on MTA as shown in Table II. The compression ratios vary with the file extensions. For example, the compression ratios of mp3 and JPEG files are close to 1, whereas bmp, doc and txt files have much smaller ratios. In addition, the compression ratio varies with compression algorithms. For example, *LZO*[3], designed for real time compression, shows a lower compression ratio. Hence, other algorithms are more preferred in our case.

We chose the Nokia N810 as an experimental device and measured the transmission cost at reference data rates through WLANs in our campus. In our power measurement setup, we used a multimeter for measuring voltage over a resistance which was serially connected to the Nokia N810. The Nokia N810 and the resistance were powered by an external DC power supply instead of batteries. The power consumption P was then calculated as $P = (U - U_R) \times U_R / R$, where U is the voltage of the DC power supply, U_R is the reading from the multimeter, and R is the resistance value. The measurement

[3] LZO website: http://www.oberhumer.com/opensource/lzo/

TABLE IV. COMPARISON OF ENERGY CONSUMPTION AND DURATION WITH/WITHOUT COMPRESSION (FILE SIZE: 2MB, COMPRESSOR: GZIP)

File Extension/Type	*With compression*		*Without Compression*		*ce*
	Energy (J)	*Duration (s)*	*Energy (J)*	*Duration (s)*	
.doc	9.61	7.0	18.31	11.8	6.90
.bmp	5.86	5.4	15.74	9.7	2.67
.pdf	25.55	22.8	28.45	23.0	1.03
.txt	13.80	12.2	18.97	13.0	2.68
Binary data	12.8	11	17.57	11.8	2.68

*ce: compression effectiveness defined in Section 3.2.

results listed in Table III were pre-stored on the proxy server together with the compression ratio and the energy model of decompression (5).

B. Performance Evaluation

We evaluated the energy-aware e-mail service by measuring the energy savings gained from adaptations. We adopted the same power measurement settings as the transmission cost at reference rates. The e-mail client was then running on the Nokia N810. We selected attachments of five different types shown in Table IV and measured the power consumption during data transmission and decompression. In Table IV, the duration of the cases with compression includes the transmission and decompression durations measured on the mobile client. The compression effectiveness was calculated based on the predicted network throughput, compression ratio, and the energy utility of decompression and transmission.

According to the results shown in Table IV, the compression of attachments could gain significant energy savings of 10% to 60%, depending on the data types of the attachments and underlying network conditions. In addition, the compression could also save transmission time, and the total duration of retrieving attachment in spite of the decompression time.

Compression effectiveness is the basis for the decision-making on compression. It shows the potential for energy savings through adaptations but not the exact amount of energy savings. The preciseness of the predicted compression effectiveness is limited by the accuracy of network throughput prediction, the energy model of decompression and network transmission, and the compression ratios obtained from benchmark files. However, since the decision-making is based on a threshold value, the values bigger than the threshold lead to the same decision. Therefore, the influence of the preciseness of compression effectiveness prediction is limited. As our experimental results have shown significant energy savings, our estimation of compression effectiveness has proved to be reasonably accurate.

Though we focus on the scenarios in which the proxy server is not battery-powered in this paper, our model is scalable to mobile-to-mobile communications scenarios with regard to future work. In such cases, the sender can play the role of proxy server. The energy models described in Section 3.2 could be updated by including the energy consumption of sending and compressing data.

The compression ratio that can be achieved varies with the data type. Certain types of files, such as PDF documents, may contain different types of data objects like images and text blocks. Hence, the prediction of compression ratio can be further refined by being based on the statistics of the data objects included in the file.

The prediction of network SNR [11] could be integrated into the prediction of network throughput, since this is influenced by the SNR value especially when the value is low. In addition, compression-based adaptations together with traffic shaping would be a practice worthy of consideration in the near future.

V. CONCLUSION

In this paper we have proposed a proxy-based energy adaptation framework which utilizes lossless data compression. Our framework aims at saving energy on mobile devices when receiving data from wireless networks. To improve the compression effectiveness, our framework takes into account the data characteristics, network transmission environment, as well as the context information of the mobile client during adaptation. The above framework has shown a gain in significant energy savings through this case study of an energy-aware e-mail service.

ACKNOWLEDGMENT

We thank Aldara De Castro Baez for programming the energy-aware e-mail service. We also extend our thanks to Petri Savolainen for his valuable comments.

REFERENCES

[1] Barr, K. C. and Asanović, K.,"Energy-aware lossless data compression," ACM Trans. Comput. Syst. vol. 24, no. 3, pp. 250-291. August 2006.

[2] Maddah, R. and Sharafeddine, S. "Energy-Aware Adaptive Compression Scheme for Mobile-to-Mobile Communications," 10th IEEE Int'l Symp. on Spread Spectrum Techniques and Applications(ISSSTA'08), pp.688-691, August 2008.

[3] Wiltten, I.H., Neal, R.M., and Cleary, J.G. "Arithmetic coding for data compression", Commun. ACM. Vol. 30, no. 6, pp.520-540. June 1987.

[4] Krintz, C. and Calder, B. "Reducing Delay with Dynamic Selection of Compression Formats," 10th IEEE Int'l Symp. on High Performance Distributed Computing(HPDC'01). pp. 266. August 2001.

[5] Krintz, C. and Sucu, S. "Adaptive On-the-Fly Compression." IEEE Trans. Parallel Distrib. Syst. vol.17,no. 1, pp. 15-24. January 2006.

[6] Mohapatra, S.; Dutt, N.; Nicolau, A.; Venkatasubramanian, N., "DYNAMO: A Cross-Layer Framework for End-to-End QoS and Energy Optimization in Mobile Handheld Devices," IEEE Journal on Selected Areas in Communications, vol.25, no.4, pp.722-737, May 2007

[7] Tamai, M., Sun, T., Yasumoto, K., Shibata, N., and Ito, M. "Energy-aware video streaming with QoS control for portable computing devices," NOSSDAV, ACM, New York, NY, pp. 68-73. June 2004.

[8] Xu, R., Li, Z., Wang, C. and Ni, P. "Impact of Data Compression on Energy Consumption of Wireless-Networked Handheld Devices," ICDCS, pp.302–311. Rhold Island, USA, May 2003.

[9] Prabhakar B., Uysal E., EL Gamal A. "Energy-efficient Transmission over a Wireless Link via Lazy Packet Scheduling", IEEE INFOCOM. Vol.1, pp 386-394. April 2001.

[10] He, Q., Dovrolis, C., and Ammar, M, "On the predictability of large transfer TCP throughput," Comput. Netw. Vol. 51, 14, pp. 3959-3977, October 2007

[11] Sri Kalyanaraman R., Xiao Y., Ylä-jääski A. "Network Prediction for Adaptive Mobile Application", UBICOMM, IEEE, October 2009.

[12] Korhonen, J. and Wang, Y. "Power-efficient streaming for mobile terminals", NOSSDAV, ACM, New York, NY, 39-44.June 2005.