The Hitchhiker's Guide to Choosing the Compression Algorithm for your Smart Meter Data

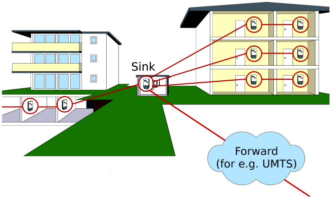
Martin Ringwelski, Christian Renner, Andreas Reinhardt (TUD),
Andreas Weigel, Volker Turau

ENERGYCON 2012
12th September, 2012

Institute of Telematics
Hamburg University of Technology
TUHH

Motivation

- Emerging renewable energy resources
- Grids to be relieved by tariffs for the current supply
- Smart Metering helps the customer
    - Shifting tasks to times with cheaper tariffs
    - Identifying heavy electricity consumers
- Meter data needs to be transmitted, but
    - Bad connectivity for GSM/UMTS in cellars
    - Powerline communication is not available in all areas
- Wireless mesh network
- Hop-to-Hop data transport



Motivation

- Pro:
  - Wireless sensor nodes are cheap
- Contra:
  - Low Bandwidth
  - Bottleneck near Sink
- Reducing network load allows more nodes in network
- Data formats are regulated and mandatory

$\Rightarrow$ Lossless data compression is needed

Smart Meter Data

ASCII-coded (EN-62056-21)

```
F.F(00000000)
0.0.0(00617827)
0.0.1(00000000)
0.9.2(1070813)
1.8.1(00000247*kWh)
1.8.1*04(00000172*kWh)
1.8.1*03(00000103*kWh)
C.51.6*03(0000000000000)
C.51.7(00000000)
C.77.2(0A78422F69654D2D4952481FBF16)
C.86.0(20001010)
31.25(0.008*A)
51.25(0.009*A)
71.25(0.045*A)
32.25(0.66*V)
52.25(0.57*V)
72.25(225.04*V)
1.35.0.01(0.000*kW)
P.01(0080213003000)(00000000)(15)(1)(1.5)(kW)
(00.00)
(00.02)
(00.01)
(02.00)
(00.01)
```

Motivation

- Trade-off: Compression gain $\Leftrightarrow$ Hardware requirements
- Sensor nodes have limited hardware resources
  (RAM, ROM, speed)
- Modern algorithms concentrate on compression gain

Question:

Which compression algorithm gives best results for the recorded data by using as few resources as possible?

What we did:

- Improve compression algorithms for wireless sensor nodes
- Analyze the strengths and weaknesses

Agenda

1. Introduction
2. Compression Algorithms
3. Evaluation
4. Conclusion

Compression Types

- **Entropy Coding:** ▷ More
  Use variable Bit-length codes for symbols, depending on their probabilities. e.g.: Huffman-, Arithmetic-, Range-Encoding
- **Wordbook Compression:** ▷ More
  Recognize repetitions of strings and use references to encode them. e.g.: LZ77, LZ78
- **Block-sorting:** ▷ More
  Rearrange the symbols in a way that they are easier to compress afterwards. e.g.: Burrows-Wheeler-Transformation, Move-To-Front

Reference Compression Algorithms

- Lempel Ziv Storer Szymanski (LZSS)
  Dictionary compression
- Lempel Ziv Welch (LZW)
  Dictionary compression (patent expired, used in GIF)
- DEFLATE (ZLIB, GZIP)
  Dictionary compression with Huffman Encoding (used in HTTP, PDF, ...)
- BZIP2
  Block sorting with Range-encoding
- Lempel Ziv Markov Chain Algorithm (LZMA)
  Dictionary compression with Reference History and subsequent Range-encoding

Entropy-based Compression

Adaptive Trimmed Huffman Coding (ATH)

▷ More

- Developed for energy-constrained wireless sensor nodes
- Adaptive Entropy Coding Scheme
- Huffman tree is trimmed to reduce memory consumption
- Prefix determines the encoding of next symbol

Adaptive Markov Chain Huffman Coding (AMCH)

▷ More

- Uses probabilities of successive symbols
- Each symbol has Huffman tree of following symbols
- Trees are built during compression
- Each tree has Escape symbol for not yet encountered (NYE) symbols

Wordbook-based Compression

tiny Lempel Ziv Markov Chain Algorithm (tLZMA)

- Adaption of the LZMA Scheme
- History window constrained to 128 Byte
- No Range-encoding step

Lempel Ziv Markov Chain Huffman Coding (LZMH)

- Combination of tLZMA and ATH
- Dictionary compression with 128 Byte History
- Compression of symbols with ATH method

Evaluation Methodology

Data

- 3 500 ASCII-coded (EN-62056-21) datasets
  - From real smart meter installation
  - Size range from 76 to 3 100 Byte
- 95 power consumption measures of household devices
  - Sampled about once per second
  - Data in binary format
  - Sizes between 18.8 and 171.0 KByte

Methodology and Metrics

- All Methods implemented in C, no use of heap memory
  ZLIB, BZ2 and LZMA implementations use heap memory
- Compression rate benchmarked on a desktop PC
- Compression rate $= 1 - \frac{\textit{compressed data size}}{\textit{uncompressed data size}}$
- Processing time benchmarked on ATmega 1281 (8 kB RAM, 7.37 MHz)
- Processing time over size $= \frac{\textit{consumed time}}{\textit{uncompressed data size}}$

Memory Consumption

Algorithm	ROM (Byte)	RAM (Byte)	
		static	stack
LZSS	544	129	19
LZW	550	12 416	16
ZLIB	27 960	2 690	ca. $>$ 1 000*
BZ2	28 332	1 564	ca. $>$ 100 000*
LZMA	34 442	110	ca. $>$ 6 000 000*
ATH	592	170	15
AMCH	1 680	1 820	21
tLZMA	992	133	27
LZMH	1 428	378	29

(*) Heap Memory

Memory Consumption

Algorithm	ROM (Byte)	RAM (Byte)	
		static	stack
LZSS	544	129	19
LZW	550	12 416	16
ZLIB	27 960	2 690	ca. > 1 000*
BZ2	28 332	1 564	ca. > 100 000*
LZMA	34 442	110	ca. > 6 000 000*
ATH	592	170	15
AMCH	1 680	1 820	21
tLZMA	992	133	27
LZMH	1 428	378	29

(*) Heap Memory

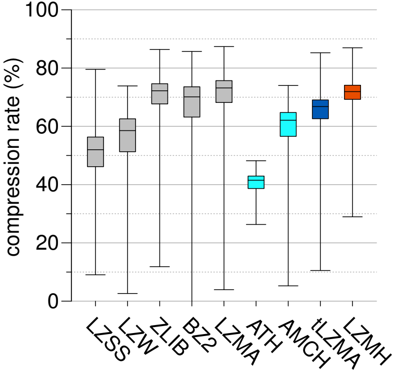
ZLIB, BZ2, LZMA and LZW use too much memory

⇒ can not be tested on

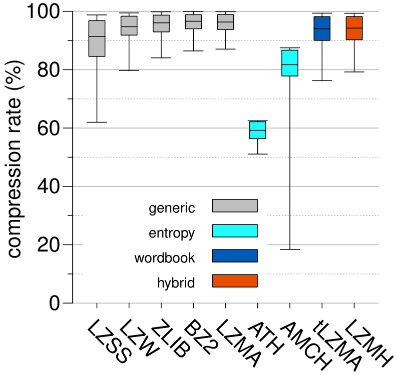
ATmega 1281

Compression Rate

ASCII encoded smart meter datasets



binary encoded daily device reports

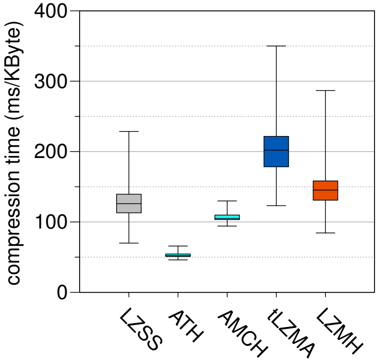


- LZMH achieves compression rates comparable to ZLIB, BZ2 and LZMA
- ATH performs good for small, but poor for bigger datasets

▷ Histogram

Processing Time

- ATH is the fastest
- ATH and AMCH are mostly unaffected by the dataset
- Other methods depend on the compressability of the datasets



▷ Histogram

Selection Guidelines

	LZSS	ATH	AMCH	tLZMA	LZMH
ROM	++	++	o	+	o
RAM	++	++	−−	++	+
Performance	o	++	+	−	o
Compression Gain	−	−−	o	+	++

Scenarios:

- Very limited resources
  - Focus on performance $\rightarrow$ ATH
  - Focus on compression $\rightarrow$ LZSS
- Limited resources, focus on compression $\rightarrow$ tLZMA
- Moderate resources, best compression $\rightarrow$ LZMH
- Good compression with predictable performance $\rightarrow$ AMCH

Resume

- Lossless data compression to reduce:
  - bandwidth
  - energy consumption
  - transmission costs
- Trade-off between compression gains, time, and resources
- ATH, LZMH, LZSS and tLZMH fulfill the resource constraints
- LZMH gives best compression results with moderate execution times and resources
- ATH is the fastest method, but has lowest compression rates

The Hitchhiker's Guide to Choosing the Compression Algorithm for your Smart Meter Data

Martin Ringwelski, Chri… TUD),
Andrea…

E…
1…

Martin Ringwelski
Research Assistant

Phone +49 / (0)40 42878 3387
e-Mail martin.ringwelski@tu-harburg.de

http://www.ti5.tu-harburg.de/staff/ringwelski

Appendix

References

- Frost & Sullivan, "Smart Meter Market – Frost & Sullivan Forecasts 109% Growth in the UK," Online: http://www.frost.com/prod/servlet/press-release.pag?docid=238393168, 2011.
- J. Vasconcelos, "Survey of Regulatory and Technological Developments Concerning Smart Metering in the European Union Electricity Market," EUI RSCAS PP 2008/01, Florence School of Regulation, 2008.
- U. Greveler, B. Justus, and D. Loehr, "Multimedia Content Identification Through Smart Meter Power Usage Profiles," in *Proceedings of the 5th International Conference on Computers, Privacy, and Data Protection (CPDP)*, 2012.
- N. Zouba, F. Brémond, and M. Thonnat, "Multisensor Fusion for Monitoring Elderly Activities at Home," in *Proceedings of the 6th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2009.
- R. C. Kryter and H. D. Haynes, "Condition Monitoring of Machinery using Motor Current Signature Analysis," in *Proceedings of the 7th Power Plant Dynamics, Control and Testing Symposium*, 2003.

References

- G. W. Hart, "Residential Energy Monitoring and Computerized Surveillance via Utility Power Flows," *IEEE Technology and Society Magazine*, vol. 8, no. 2, 1989.
- J. Ziv and A. Lempel, "A Universal Algorithm for Sequential Data Compression," *IEEE Transactions on Information Theory*, vol. 23, no. 3, pp. 337-343, 1977.
- T. A. Welch, "A Technique for High-Performance Data Compression," *IEEE Computer*, vol. 17, no. 6, pp. 8-19, 1984.
- G. J. Pottie and W. J. Kaiser, "Wireless Integrated Network Sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51-58, 2000.
- K. Barr and K. Asanović, "Energy Aware Lossless Data Compression," in *Proceedings of the 1st International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2003, pp. 231-244.
- N. Tsiftes, A. Dunkels, and T. Voigt, "Efficient Sensor Network Reprogramming through Compression of Executable Modules," in *Proceedings of the 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 2008, pp. 359-367.

References

- C. M. Sadler and M. Martonosi, "Data Compression Algorithms for Energy-Constrained Devices in Delay Tolerant Networks," in *Proceedings of the 4th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2006, pp. 265-278.
- T. Schoellhammer, B. Greenstein, E. Osterweil, M. Wimbrow, and D. Estrin, "Lightweight Temporal Compression of Microclimate Datasets," in *Proceedings of the 29th IEEE Conference on Local Computer Networks (LCN)*, 2004, pp. 516-524.
- E.-O. Blass, L. Tiede, and M. Zitterbart, "An Energy-Efficient and Reliable Mechanism for Data Transport in Wireless Sensor Networks," in *Proceedings of the 3rd International Conference on Networked Sensing Systems (INSS)*, 2006, pp. 211-216.
- A. Reinhardt, "Designing Sensor Networks for Smart Spaces – Unified Interfacing and Energy-Efficient Communication between Wireless Sensor and Actuator Nodes," Ph.D. dissertation, Technische Universität Darmstadt, Multimedia Communications Lab, 2011.

References

- J. Z. Kolter and M. J. Johnson, "REDD: A Public Data Set for Energy Disaggregation Research," in *Proceedings of the SustKDD Workshop on Data Mining Applications in Sustainability*, 2011.
- A. Reinhardt, D. Christin, M. Hollick, J. Schmitt, P. Mogre, and R. Steinmetz, "Trimming the Tree: Tailoring Adaptive Huffman Coding to Wireless Sensor Networks," in *Proceedings of the 7th European Conference on Wireless Sensor Networks (EWSN '10)*, no. LNCS 5970, Coimbra, Portugal, Feb. 2010.
- D. Huffman, "A Method for the Construction of Minimum-Redundancy Codes," *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098-1101, 1952.
- J. S. Vitter, "Design and Analysis of Dynamic Huffman Codes," *Journal of the Association for Computing Machinery*, vol. 34, no. 4, pp. 825-845, 1987.
- D. Salomon and G. Motta, *Handbook of Data Compression*, 5th ed. Springer, 2010.

Entropy

- Measure for the average self-information and the information density of a code system
- n: Number of symbols, p_i: Probability of a symbol

  $$H = -\sum_{i=1}^{n} p_i \cdot \log_2 p_i$$
- The entropy is the average number of needed bits to encode one symbol in a message

▷ Back

Move-To-Front Example

Symbol	Sequence	Alphabet
b	1	abcdefghijklmnopqrstuvwxyz
a	1	bacdefghijklmnopqrstuvwxyz
n	13	abcdefghijklmnopqrstuvwxyz
a	1	nabcdefghijklmopqrstuvwxyz
n	1	anbcdefghijklmopqrstuvwxyz
a	1	nabcdefghijklmopqrstuvwxyz

▷ Back

LZSS Example

```
Hello you there, yea you there, yea you there,
Hello!
```

Becomes:

```
Hello you there, yea(5,11)(18,15)(0,5)!
```

The Numbers in brackets determine the offset and length of the previous occurrence of the following string.

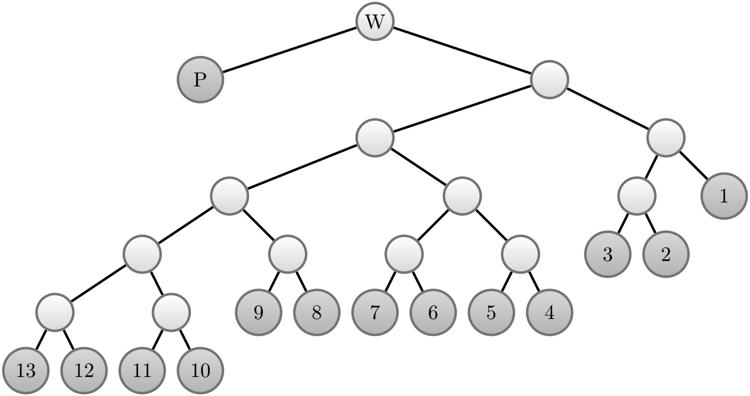
▷ Back

Range-encoding Example

Message: AABA$
Intervals: A: 0 to 0,6 - B: 0,6 to 0,8 - $: 0,8 to 1
A - 0 to 0,6
A - 0 to 0,36
B - 0,216 to 0,288
A - 0,216 to 0,2591
$ - 0,25046 to 0,2591
$\Rightarrow$ 0,251 is the shortest number that lies in the end interval and thereby encodes the message AABA$.

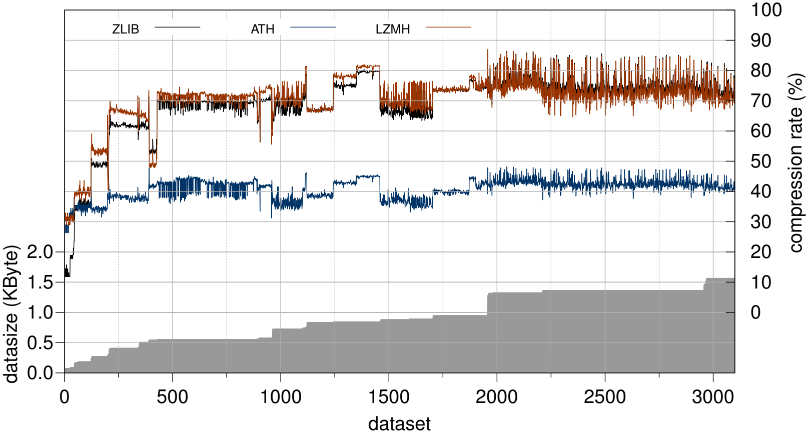
▷ Back

The used ATH tree



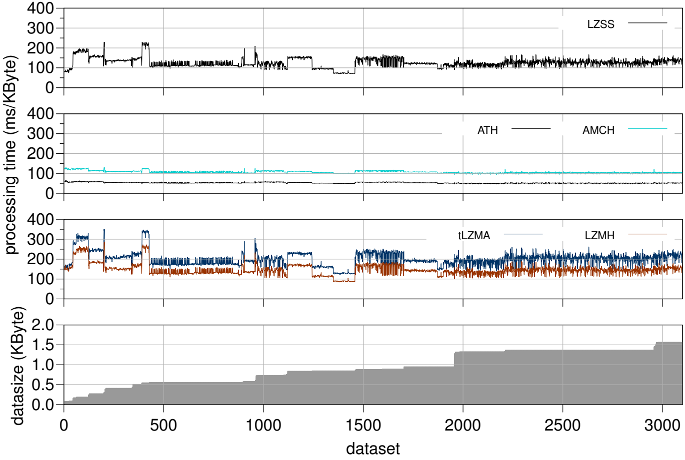
▷ Back

Histogram



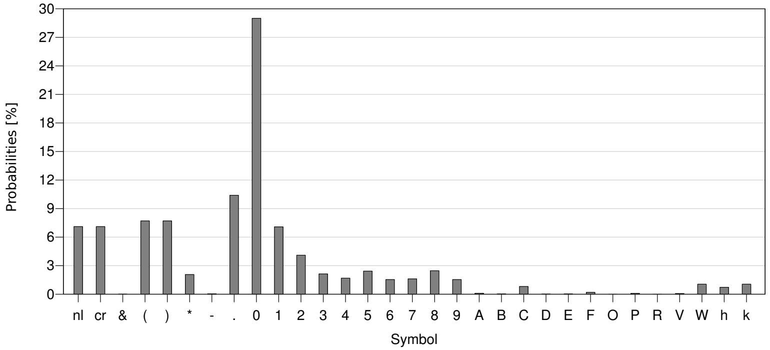
▷ Back

Histogram of Times

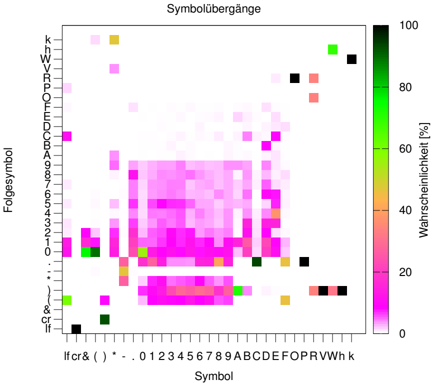


▷ Back

Symbol Histogram



Symbol-transition Heatmap



▷ Back