ALBERT-LUDWIGS-UNIVERSITÄT FREIBURG

Institut für Informatik

Efficient Aggregation of encrypted data in Wireless Sensor Networks

Claude Castelluccia, Einar Mykletun, Gene Tsudnik

Refik Hadzialic

Rechnernetze und Telematik
Seminar: Ad-Hoc Networks

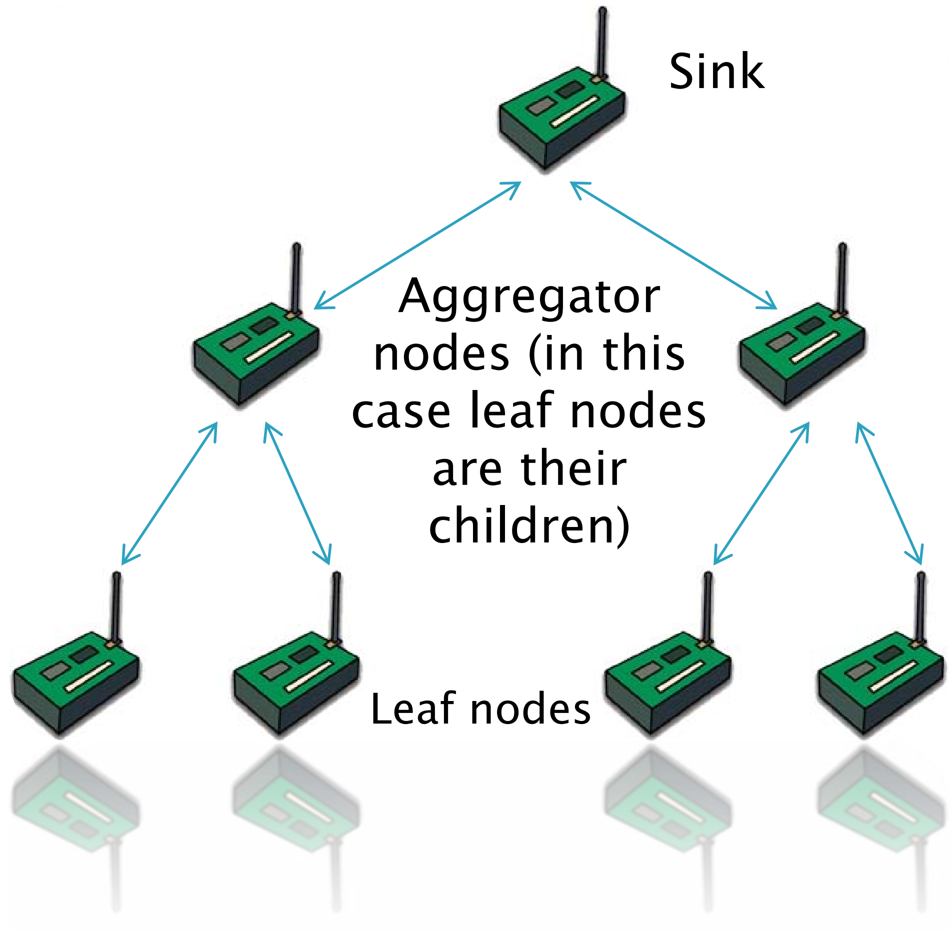
Prof. Christian Schindelhauer, Arne Vater

Agenda of the Presentation

- Introduction
- Sensor nodes
- Aggregation of data
- Protection
  - Encryption
  - Homomorphic property
  - Key stream generation and stream cipher
- Computation of Average and Variance
- Analysis
- Results
- Improvements

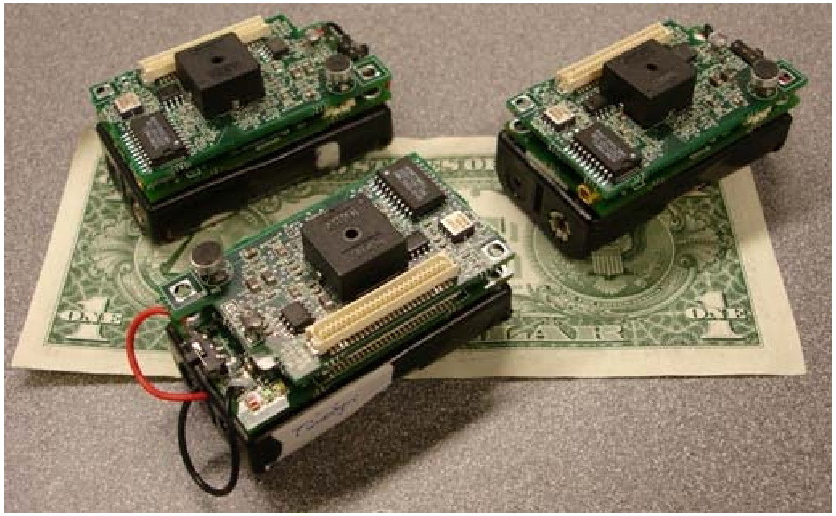
Introduction to the topic

- What is an WSN?
- Goals of WSNs
  - Monitoring of the environment
- Limitations of Nodes
  - Limited computation power
  - Battery powered
  - RF (ISM Band) power limitation

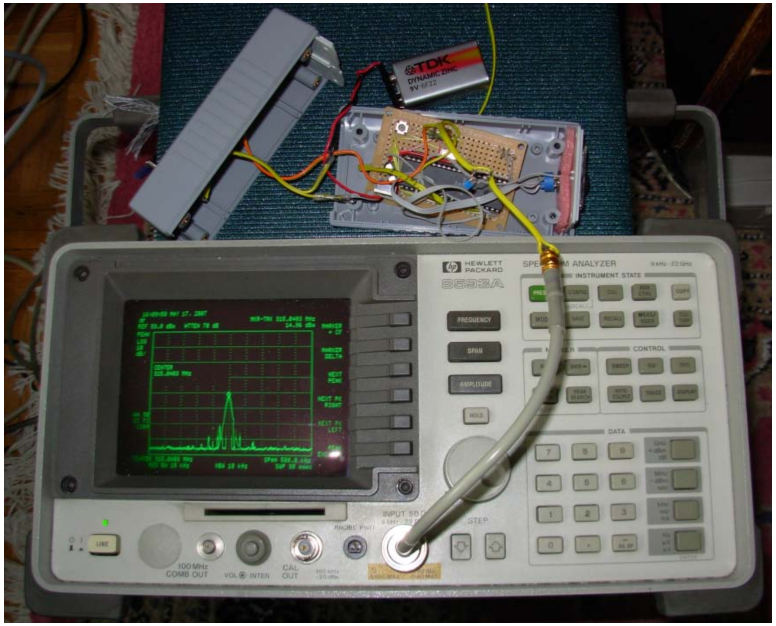


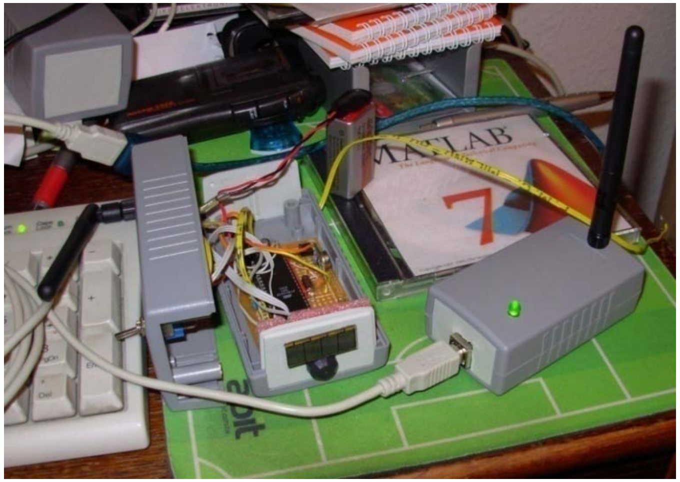
Node picture taken from prof. Christian Schindelhauer; Presentation: 08-A-WSN-Einfuehrung; page#: 2

Typical sensor nodes



MICA2Mote

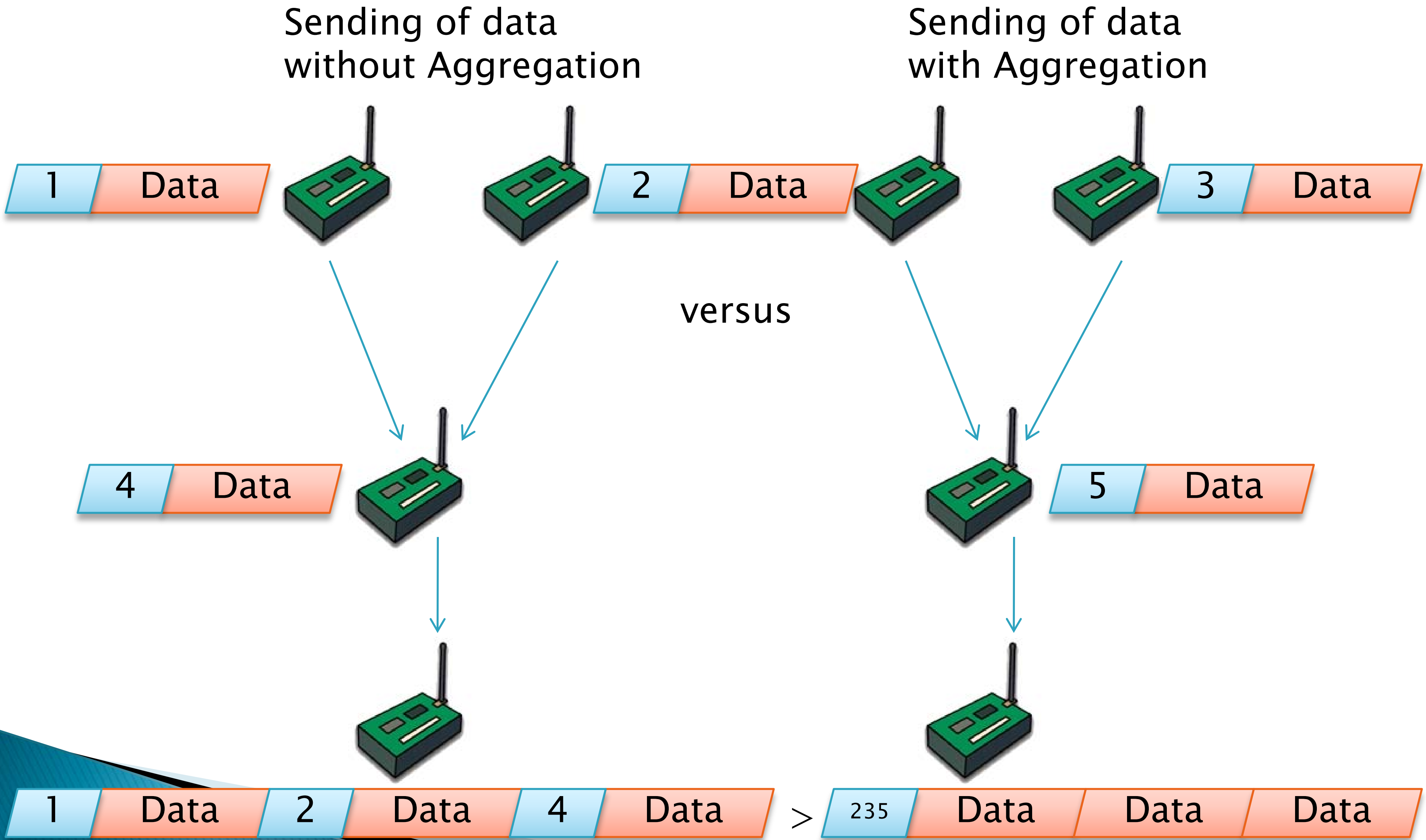




My own developed Sensor nodes
for a Wireless Alarm System

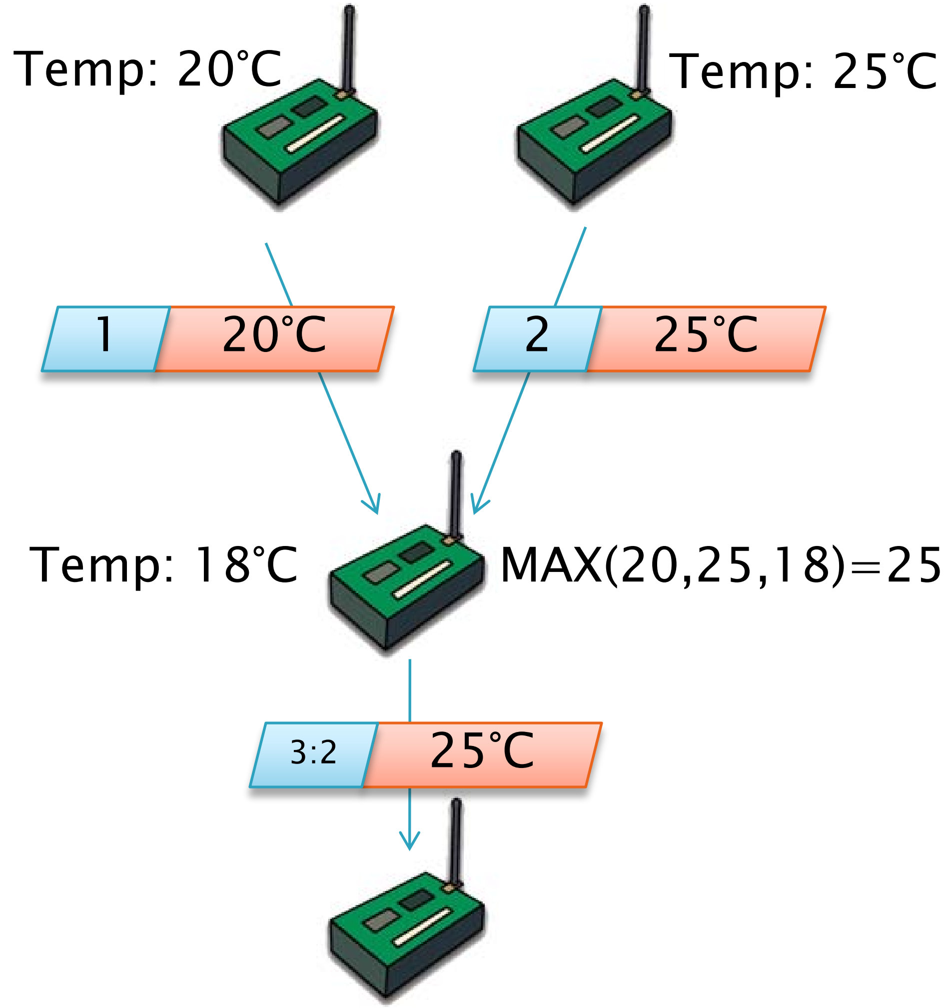
MICA2Mote picture taken from: http://www.eecs.berkeley.edu/~watteyne/index.html

Aggregation of Data



Aggregation of data

- In most cases we need only the max/min or the mean value.



Idea taken from prof. Christian Schindelhauer; Presentation: 09-B-WSN-Aggr-1; Page #:5

Protection of data

- We need to protect our data from listeners (e.g. a competitor company who wants to use our sensor data for their products)
- Predator MQ-1
- Solution: Encrypt data
- End-to-end encryption

Source of the image: http://www.eteamz.com/10ULadyRedDevils/

Aggregation + Encryption

- Aggregation and encryption problems?
- Aggregate the data without giving any node the privilege of knowing what is inside in the packet of its child?

Aggregation + Encryption

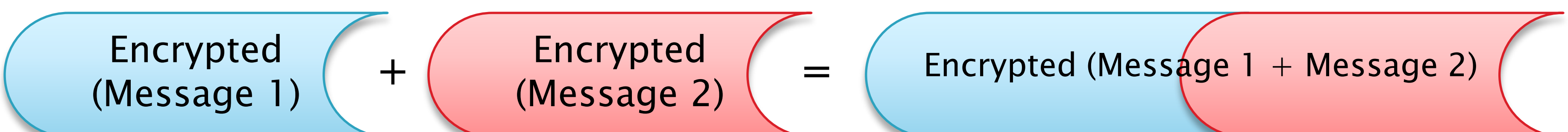
- Aggregation and encryption problems?
- Aggregate the data without giving any node the privilege of knowing what is inside in the packet of its child?
- YES!
- By using the property of homomorphic encryption algorithms.

Property of homomorphic encryption algorithms

- What is homomorphic encryption?

- Voting system!

- $\xi(X+Y)=\xi(X)+\xi(Y)$, where $\xi(X)$ denotes encryption of some value X.



Encryption algorithm

- Encryption:

$$c_i = Enc(m_i, k_i, M) = m_i + k_i (mod M)$$

- Decryption:

$$Dec(C, K, M) = C - K (mod M)$$

where:
m – message
k – secret key
M – big modulo number
C – sum of decrypted messages ($C=c_1+c_2+\ldots+c_n$)
c – decrypted message (cipher)
K– sum of all secret keys ($K=k_1+k_2+\ldots+k_n$)

Conditions for the encryption

- Discrete logarithm scheme

$$c_1 \cdot c_2 = Enc_k(m_1 + m_2)$$

- This property holds as long as:

$$M = 2^{\lceil \log_2(p*n) \rceil}$$

$$p = \max(m_i)$$

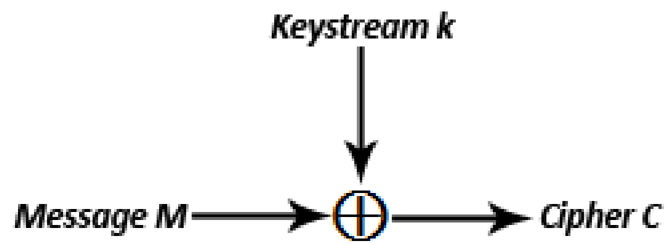
$$n - number\ of\ nodes$$

- k – generated randomly using a stream cipher

Stream cipher

- e.g. plaintext message byte M=145
- e.g. keystream byte k=234

- C=M⊕k
- C=123



- RC4 can encrypt ~ 1MByte/s on 33 MHz, ATMEL MCU works on around 16 MHz

Source of the speed information: ftp://ftp.rsasecurity.com/pub/pdfs/tr701.pdf

Computation of Average and Variance

- DSP
- Average calculation

$$\mu = \frac{1}{N}\sum_{i=0}^{N-1} x_i$$

- Variance calculation

$$V = \sum_{i=0}^{k-1} x_i^2$$

$$Var = E(x^2) - E(x)^2$$

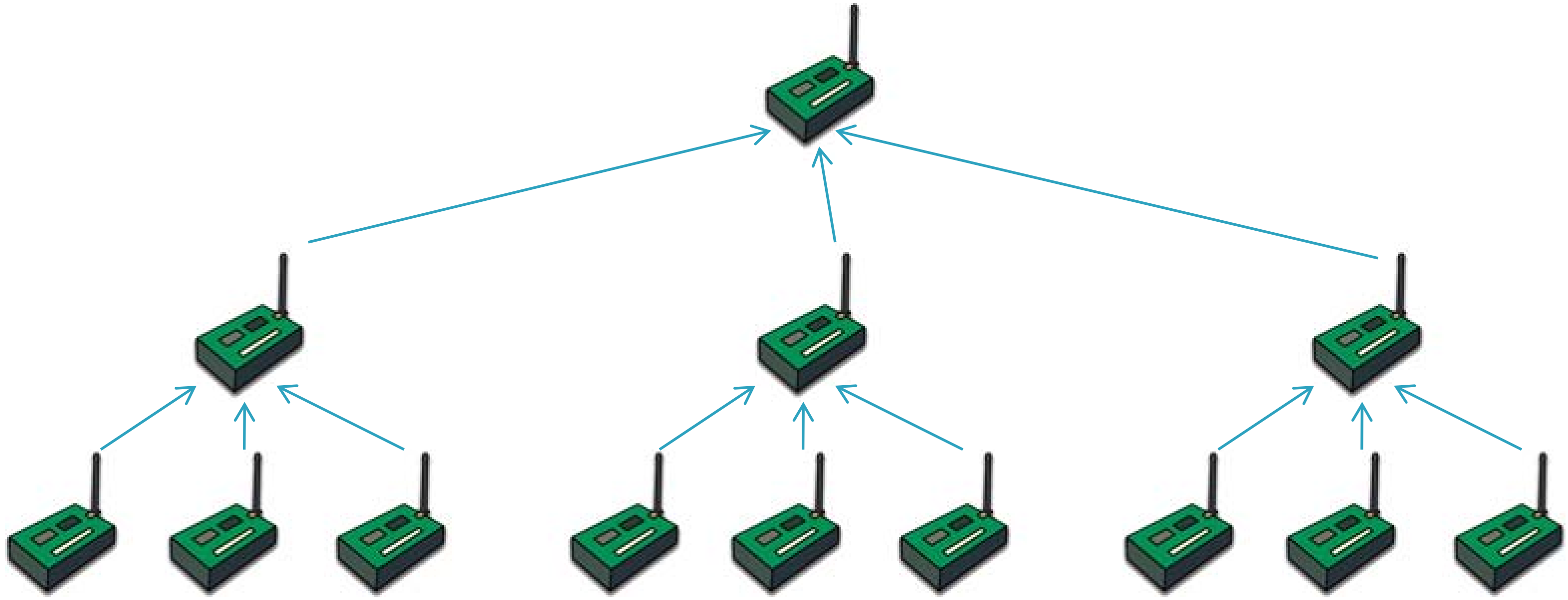
$$E(x^2) = \frac{\sum_{i=0}^{k-1} x_i^2}{k}$$

$$E(x) = \frac{\sum_{i=0}^{k-1} x_i}{k}$$

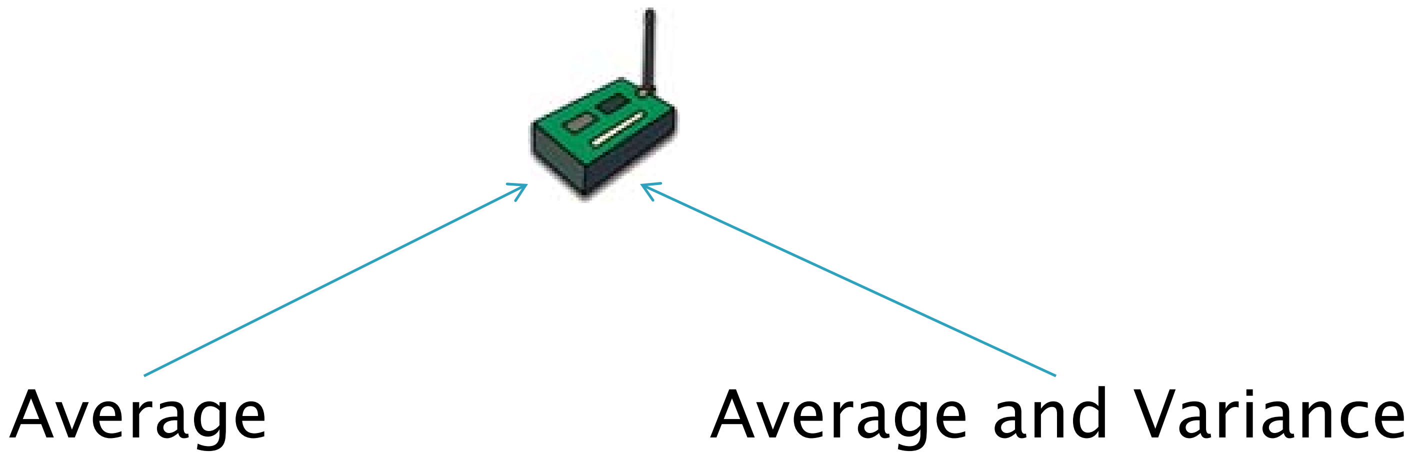
Analysis methods

- 3-ary tree

- No aggregation(No-Agg)
- Hop-by-Hop (HBH)
- Proposed Aggregation method (AGG)



Analysis methods

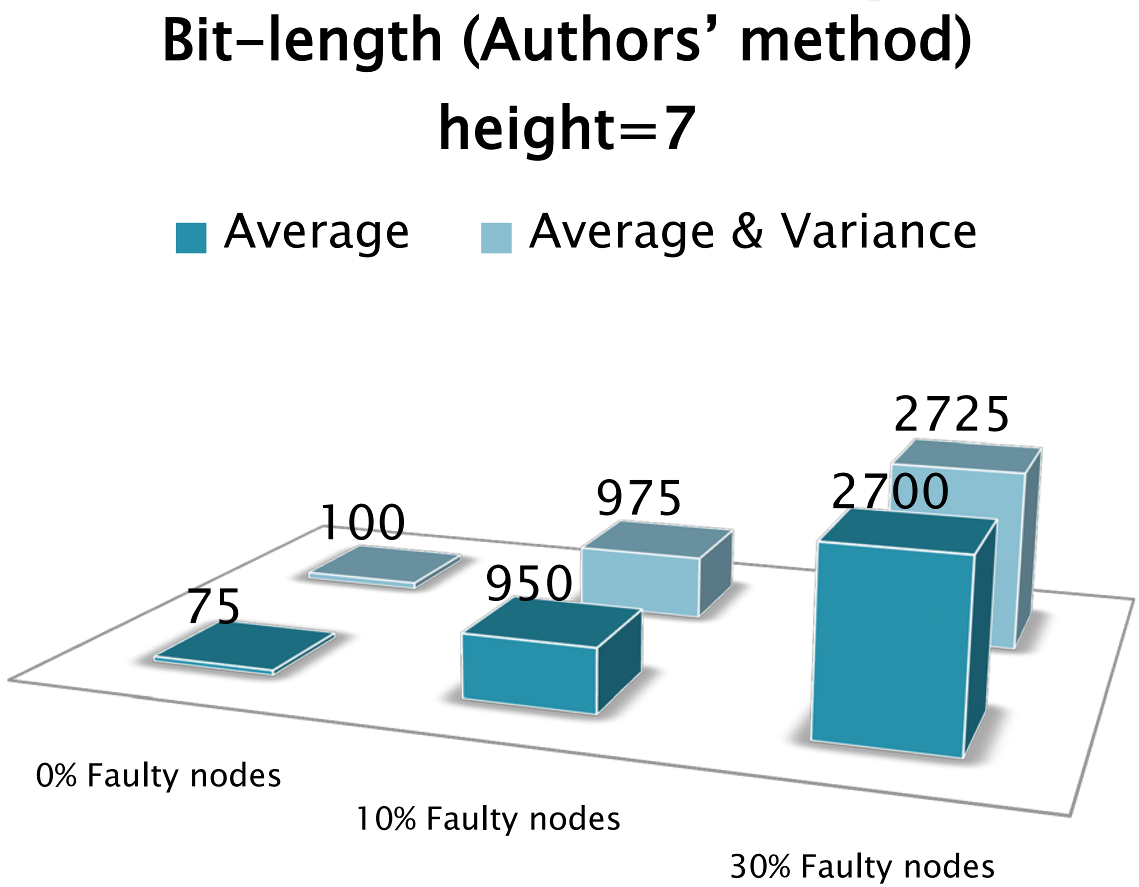


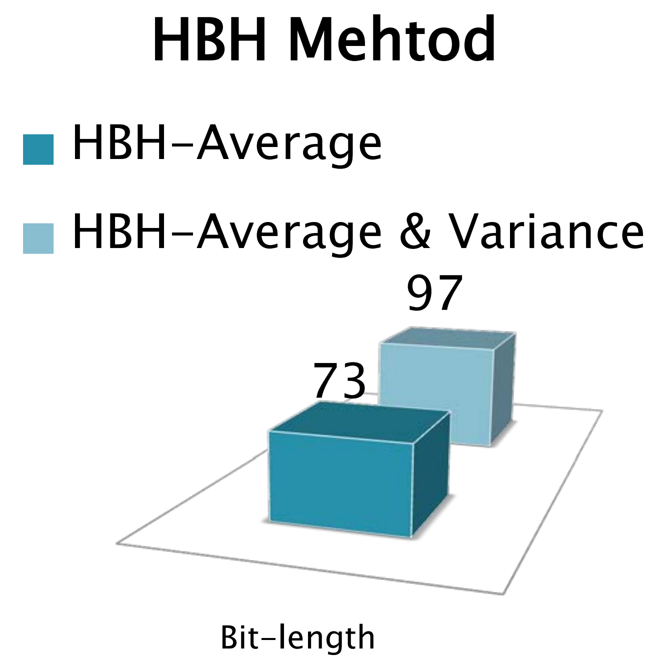
- Interests:
  - The number of bits sent per node at different levels in a 3-ary tree
  - The total number of bits transmitted throughout the WSN for 3-ary trees of various heights

Analysis

- No aggregation – safest, no bandwidth gain, nodes near the sink die asap
- HBH – best bandwidth gain, no end-to-end encryption, easy hackable, battery power used for encryption and decryption
- Authors method – bandwidth efficient, end-to-end encryption

Results – Comparison





No aggregation method would require: **68859** bits

Result analysis

$$M_A = n \cdot t$$

- Take the log of this

$$56 + log(t) + log(n) =$$
$$56 + log(128) + log(2187) =$$
$$= 56 + 7 + 12 = 75$$

where n – number of nodes

t – max(m_i)

Improvements

- Is there a way to improve the compression ratio?

Improvements

- Is there a way to improve the compression ratio?

- YES! There is a way of doing this by taking into account some known facts. BUT HOW?

Improvements – Huffman encoding

- Idea: replace frequently occurring symbols with a smaller bit representation than those that occur rarely

- e.g. average temperatures in June, in Sarajevo

Highest temp. June	Lowest temp. June
+27°C	*+14 ° C*

Idea about Huffman encoding taken from Steven Smith. *Digital Signal Processing: A Practical Guide for Engineers and Scientists.* 2002.

Improvements – Delta encoding

- Original data stream: 17 19 24 24 21
- Delta encoded: 17 +2 +5 0 −3
- pros: when sample-to-sample values deviate slowly (which is mostly the case in temperature)
- cons: won't work easily if average values are needed; requires knowing the order of sent data

Idea about Delta encoding taken from Steven Smith. *Digital Signal Processing: A Practical Guide for Engineers and Scientists.* 2002.

Conclusion

- New approach to the problem
- End-to-end encryption
- Fast (not much CPU power is used)
- Aggregation of data
- Bandwidth efficient
- Equally distributed communication load
- Strong level of security
- Can be improved more (Huffman, Delta encoding, etc.)

References

- Efficient Aggregation of encrypted data in Wireless Sensor Networks, Claude Castelluccia, Einar Mykletun, Gene Tsudnik
- Drahtlose Sensornetze: Datenaggregation, Algorithmen für drahtlose Netzwerke, Christian Schindelhauer
- Applied Cryptography, Bruce Schneier, Stream Cipher
- A Method of Homomorphic Encryption, XIANG Guang-Li
- The rest of used things was already mentioned in the footer of slides.

Thank you

- Well, that would be it. Thank you for listening. I hope you enjoyed this as much as I did.
- Questions!?!?!?

☺