

Error Correction Code for Chipcon Radio

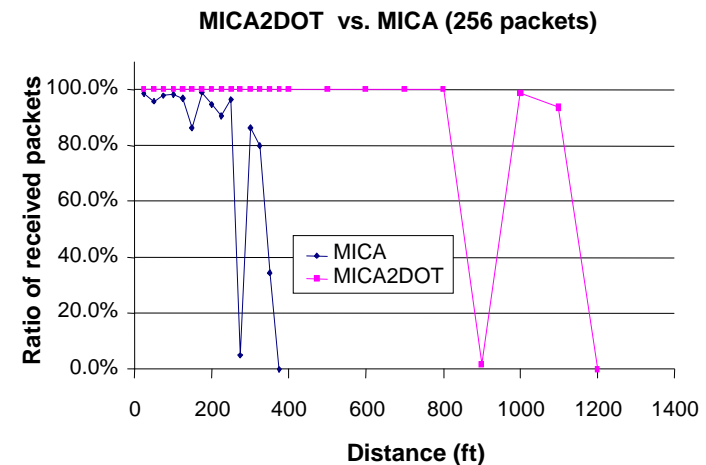
**NEST Summer Retreat 2003
June 18th, 2003**

Jaein Jeong, Cheng Tien Ee

Introduction



- A wireless sensor
 - Communicates using network stack.
 - Data bits are modulated and transmitted by radio chip.
- Chipcon Radio (CC1000)
 - Used for new platforms MICA2 / MICA2DOT.
 - Advantages over RFM radio
 - » Longer range: 900 ft vs. 300 ft
 - » Modulation: FSK vs. OOK
 - » Bit encoding: NRZ and Manchester vs. NRZ encoding
- Our goals
 - To find error characteristics
 - To develop effective error correction schemes for Chipcon radio.



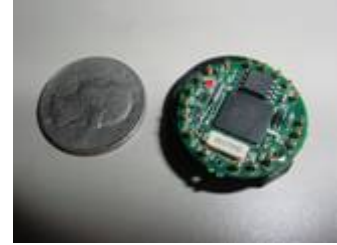
**A Mica2Dot mote
(radio chip in the middle)**

Recovering packet errors

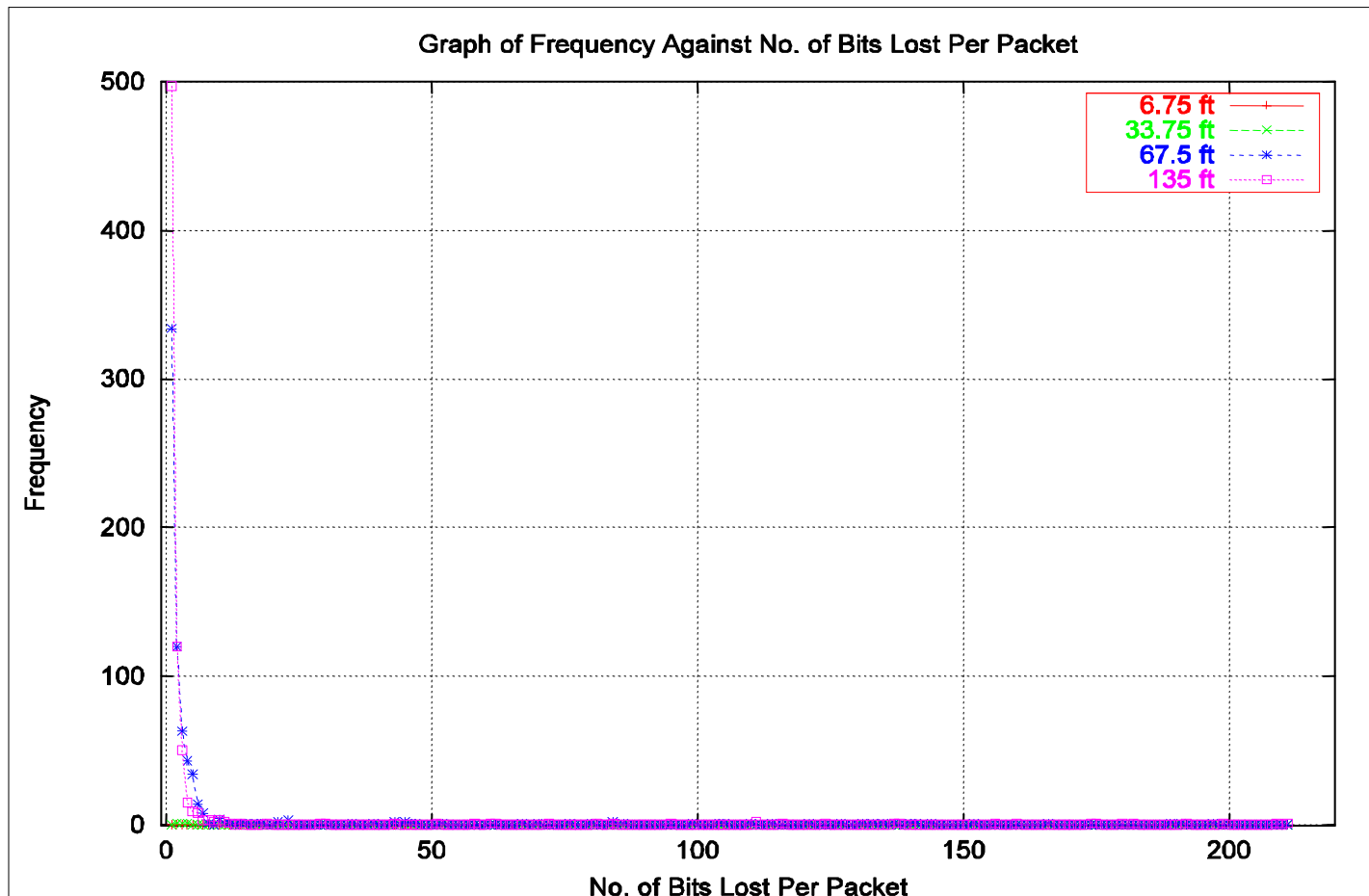


- Wireless medium can introduce packet errors.
 - We need a way to recover packets.
- Two ways of recovering packets:
 - Forward Error Correction (FEC)
 - » recipient recovers data bits using additional bits (parity).
 - Automatic Repeat Request (ARQ)
 - » recipient requests the retransmission of lost packets.
- We use error correction code (ECC) with the following observations:
 - Most corrupted packets have single or double bit errors.
 - Packet corruption rate is still higher than wired links.
 - ARQ is not suitable for broadcast communication pattern.

Preliminary Measurement



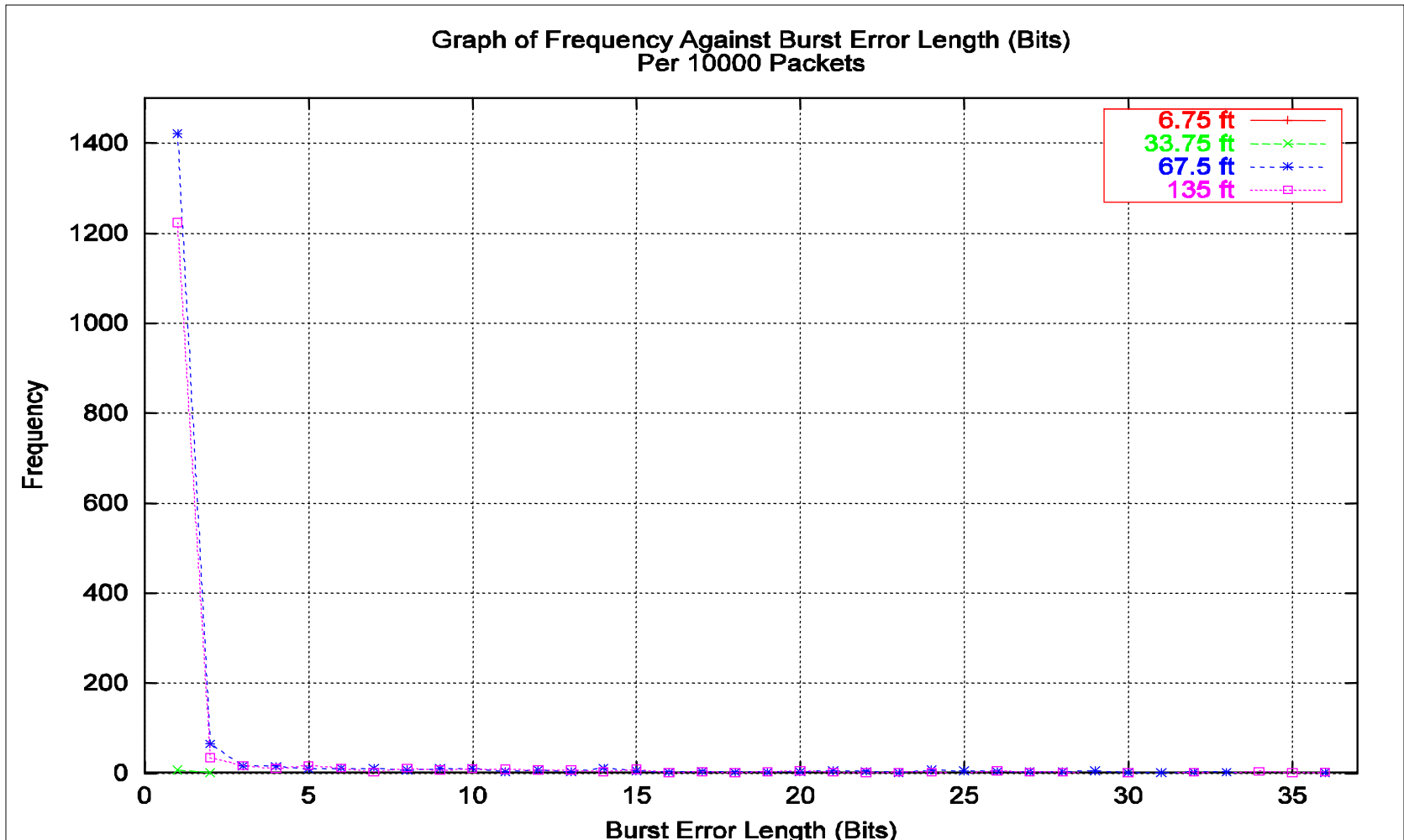
- Data was gathered outdoors (1 pkt = 36 bytes).
- Frequency of corrupted bits per packet w/o ECC



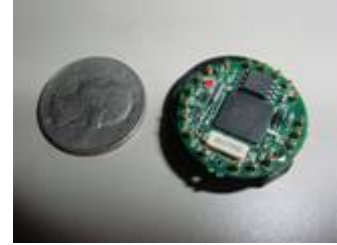
Preliminary Measurement (cont.)



- Most packets have few burst errors (< 3 bits).



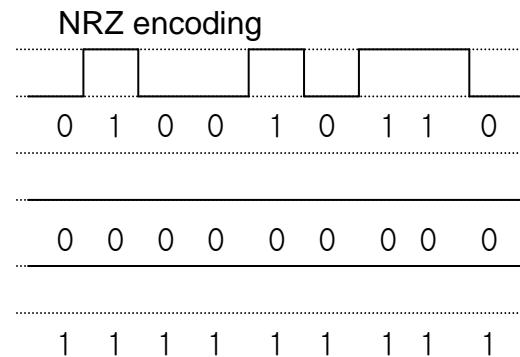
How are data bits sent?



- Bit encoding schemes specify how data bits are encoded before being sent.

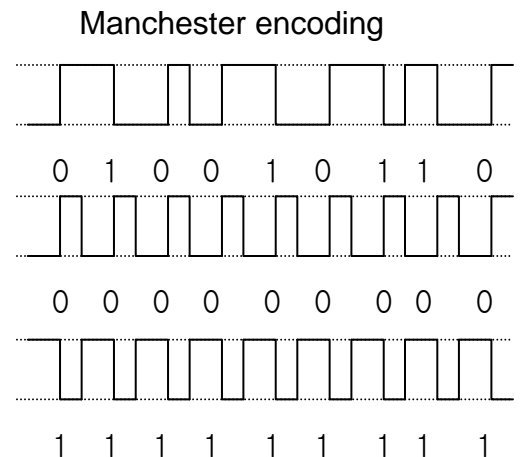
- NRZ encoding

- Encodes 0 as low and 1 as high
- Simple but needs DC balancing
- Long sequence of 0s or 1s makes the physical medium not detect the signal level correctly.

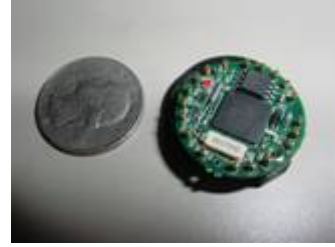


- Manchester encoding

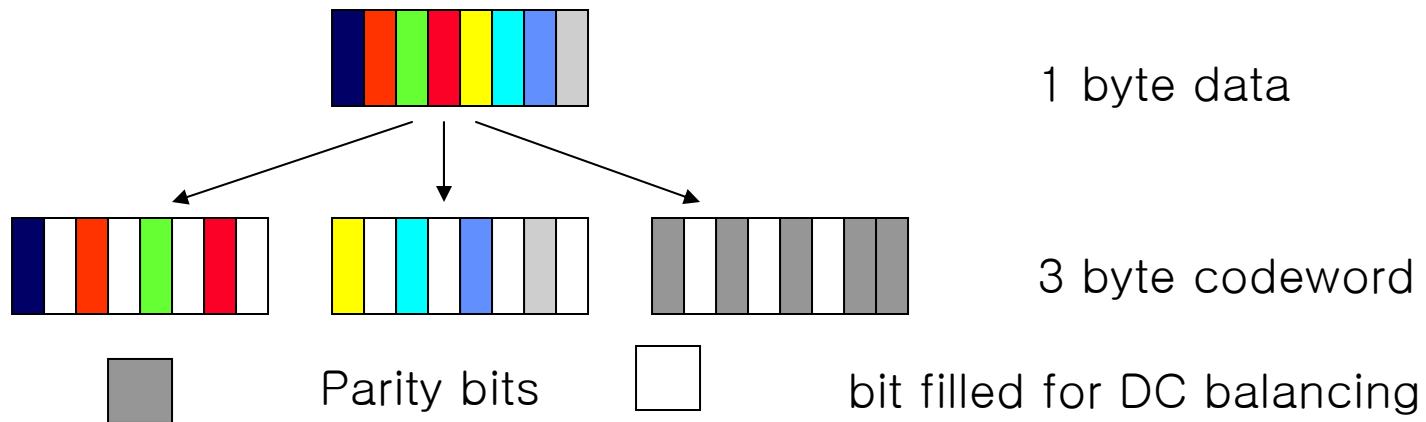
- Encodes 0 as rising edge and 1 as falling edge
- DC balanced, but utilizes only 50% of clock cycle



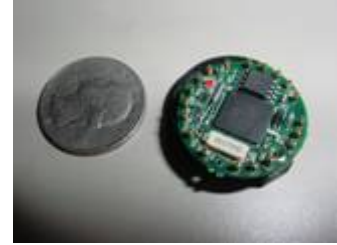
Comparison of Chipcon Radio with previous platform



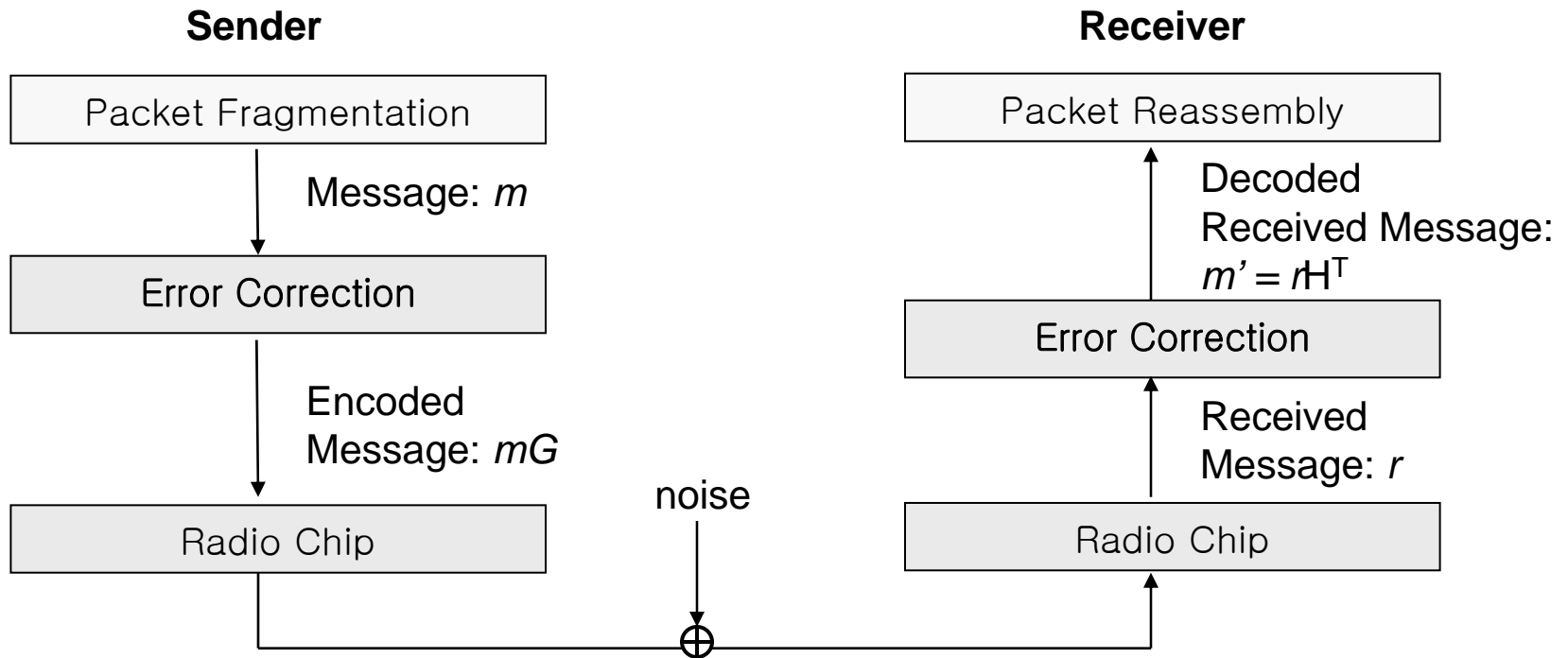
- Problems with ECC implementation for RFM radio.
 - ECC module does DC balancing besides correcting bits.
 - Still works for Chipcon, but inefficient.
 - » 3 bytes codeword for 1 byte data
(8-bit data, 5-bit parity, 11-bit bit-stuffing)



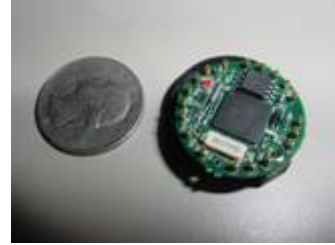
How does error correction code work?



- Data bytes are encoded using ECC before being sent.
- Received data bytes are decoded using ECC.

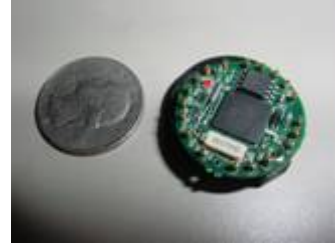


Error Correction Code: Encoding



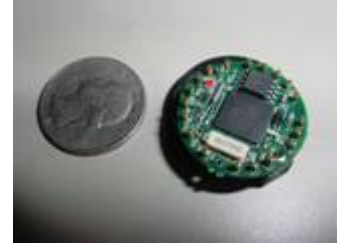
- Data bytes m can be encoded to mG using generator matrix G .
- For k -bit data, generator matrix G has the form $[I_k:C]$
- Generator matrix G has the following form:
 - $G = [I_k:C]$ for k -bit data
 - I_k : $k \times k$ identity matrix
 - C : $k \times r$ binary matrix

Error Correction Code: Decoding



- On Receiver end, syndrome s is calculated from received data r .
 - $s = rH^T = (m+e)H^T = eH^T$, e = error vector
- Parity matrix H is made from generator matrix G and of the following form:
 - $H = [C^T : I_r]$ (r : number of parity bits)
- Errors can be found by checking syndrome:
 - $s = 0 \Rightarrow$ no errors
 - Syndromes s_c for correctable errors patterns are unique and nonzero. \Rightarrow
Bit errors can be found by comparing error patterns.

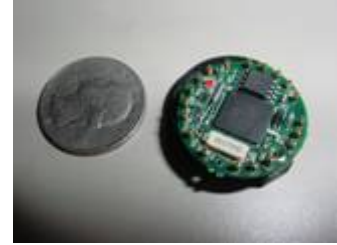
Odd-weight column Code



- Odd-weight column code can correct 1-bit errors and detect 2-bit errors (SECDED: Single Error Correction Double Error Detection).
- Odd-weight column code has odd-weight columns for H matrix.
 - r : # parity bits, k : # data bits.
 - Number of columns of H matrix should be $\geq r + k$
- For example, $k = 8$ bits data, $r = 5$

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$H = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$



Example: (16,8) Systematic QC Code

- Example:
 - Message: 0100 0010
 - Encoded message:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

- Assume bits 2 and 3 flipped

» Encoded message: $[0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0]$

» Received message: $[0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0]$

Example: (16,8) Systematic QC Code (cont.)



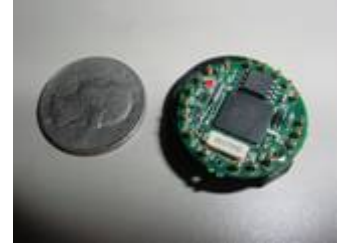
- Multiply received message by H^T

$$s = rH^T = [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0] \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} = [1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1]$$

\downarrow
 $[0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0]$

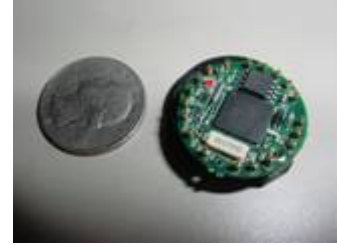
- Syndrome is exclusive-or of 2nd + 3rd rows of $H^T \Rightarrow$ 2nd and 3rd bit error

Experiments



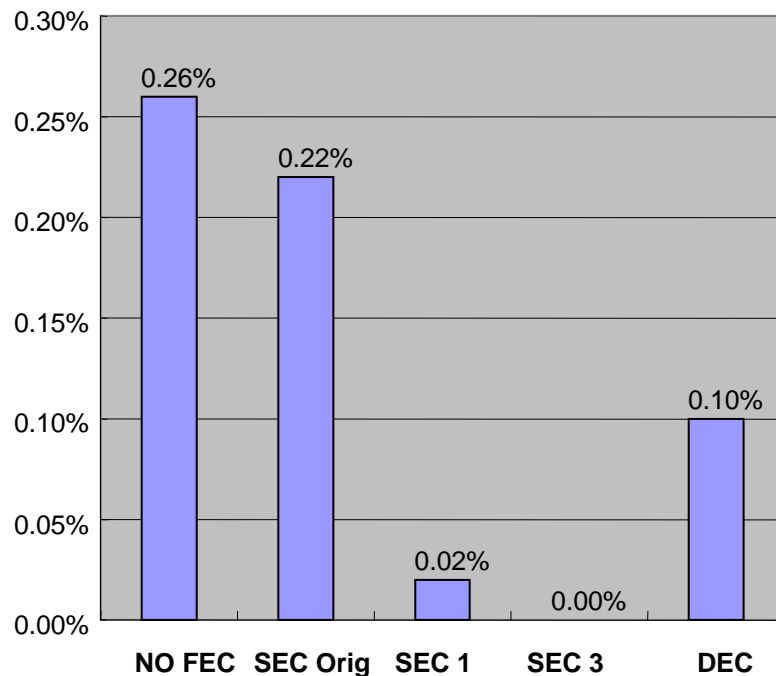
- Tested schemes:
 - No ECC
 - SECDED for RFM radio
 - SECDED (13, 8)
 - SECDED (30, 24)
 - DEC (16, 8)
- The same pair of transmitter/receiver nodes for different encoding schemes and locations
 - Avoids the effects from manufacturing deviation.
- The transmitter node sends the same packet 5000 times.
- The received data is logged in PC for analysis.

Results (Outdoors)

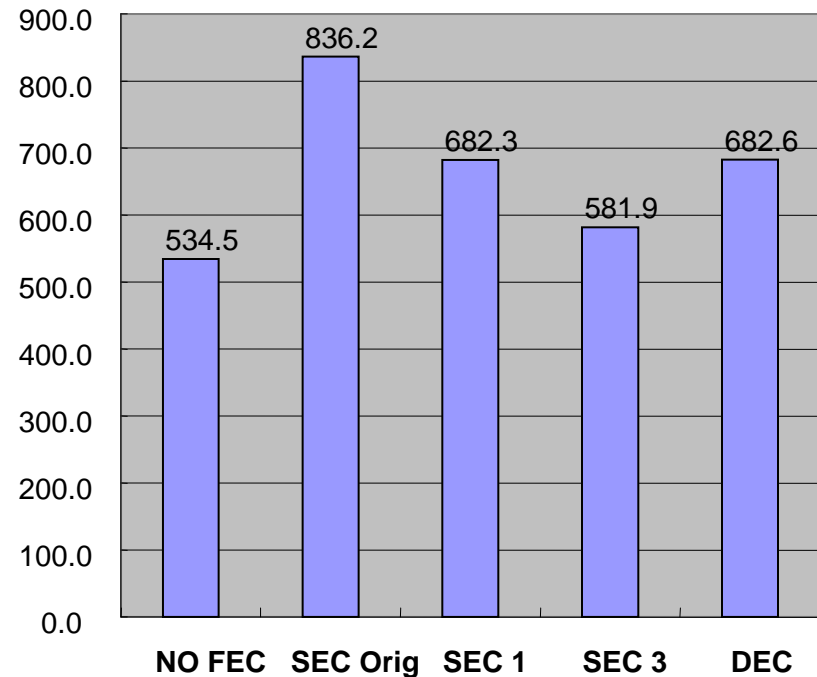


- In outdoor experiments, our ECC implementation was effective for correcting packet errors.

Packet Loss Rate: 5000 packets



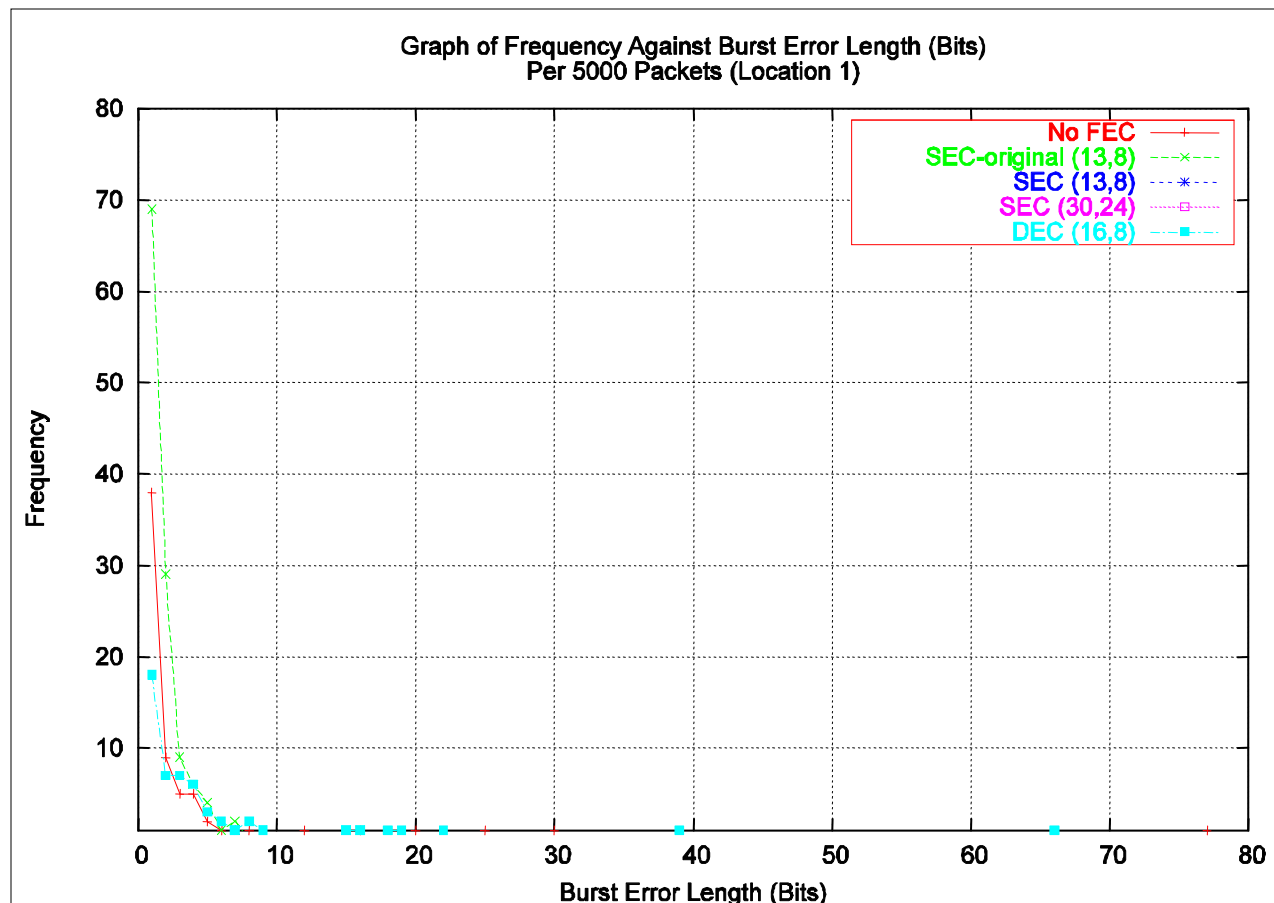
Running Time(s): 5000 packets



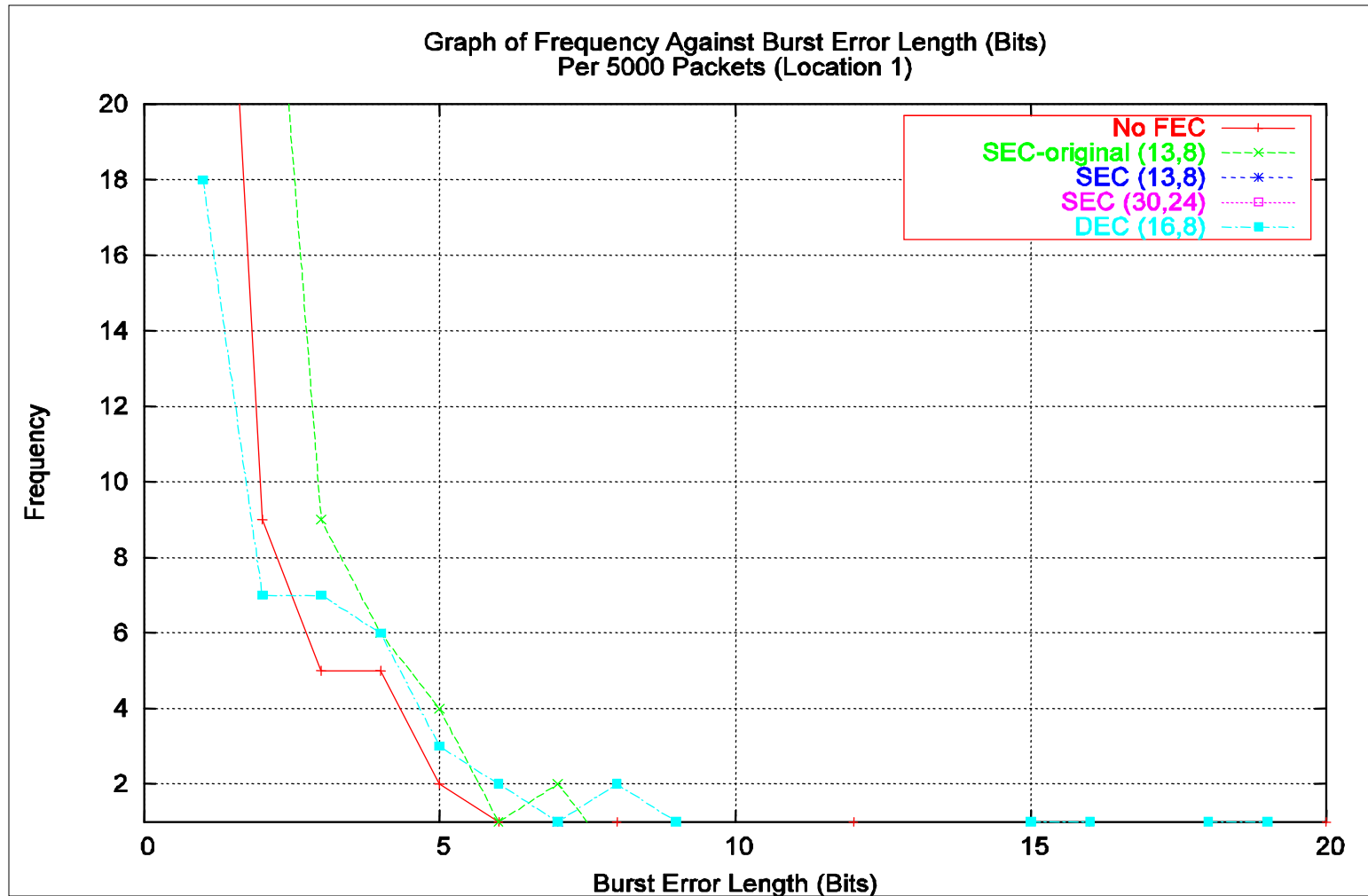
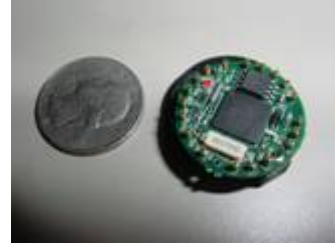
Outdoor Burst Bit Errors



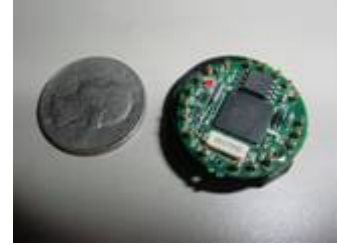
- Burst errors occurred, but rarely.
- Most errors were 1 or 2 bit errors.



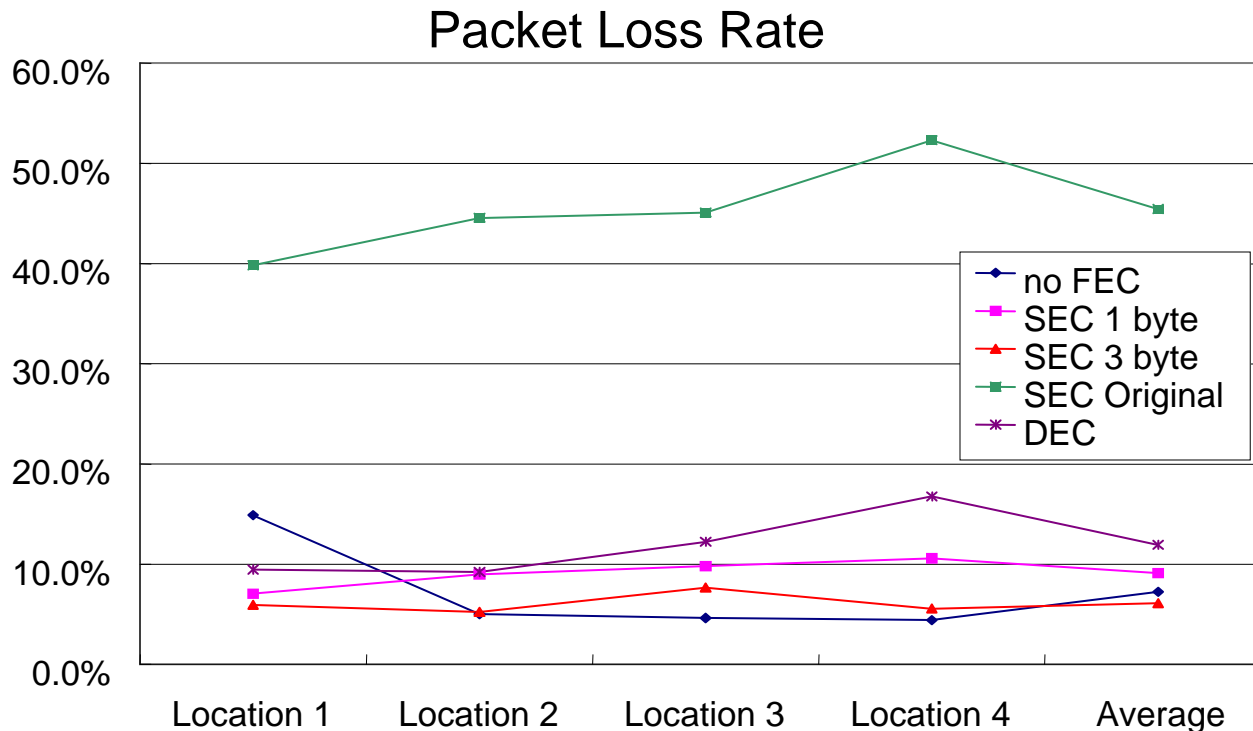
Outdoor Burst Bit Errors (cont.)



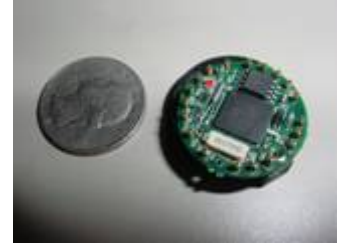
Results (Indoors)



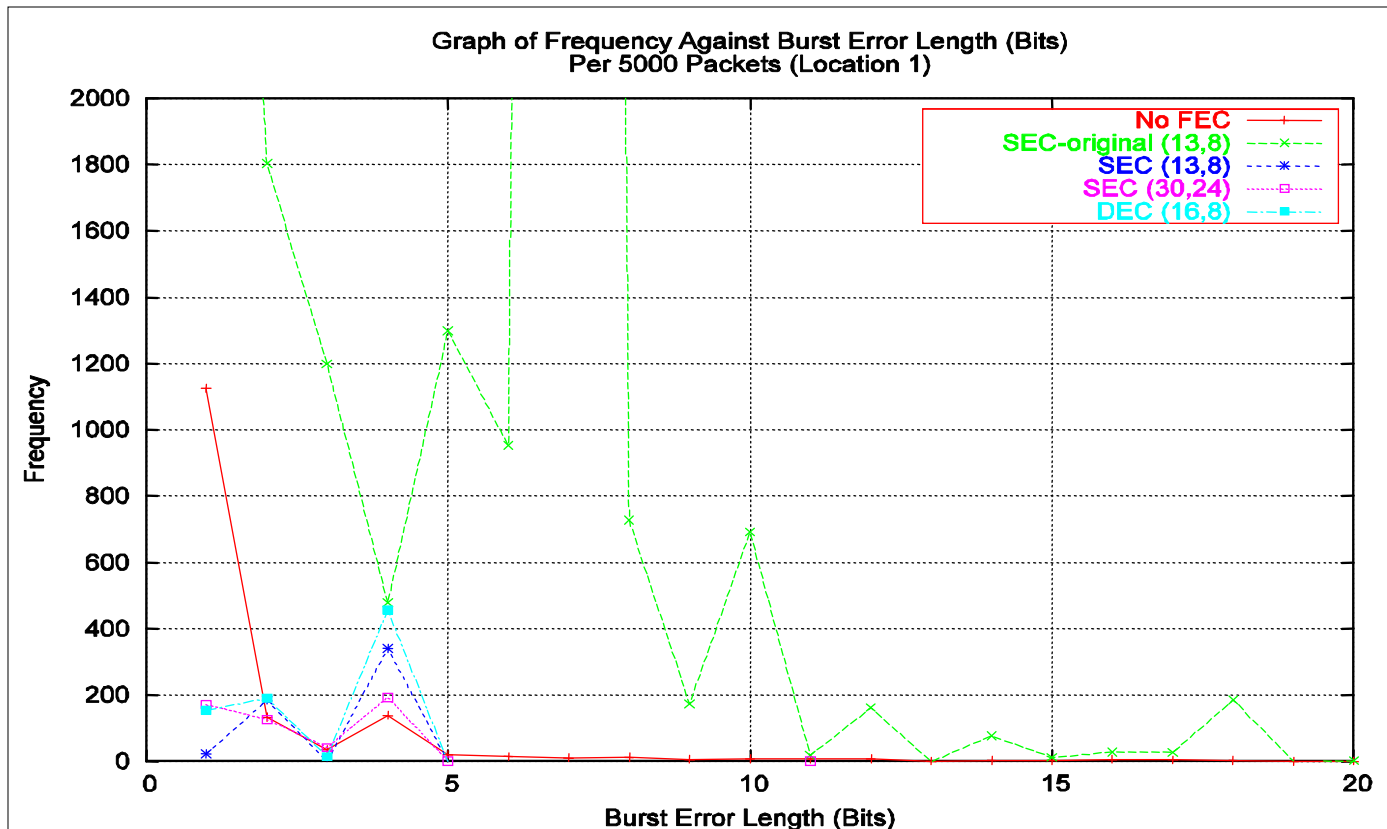
- ECC was not so effective for indoor tests.
- Rather high packet loss rate.
 - Our ECC & no FEC had 10% of packet loss.
 - SECDED for RFM performed consistently badly.



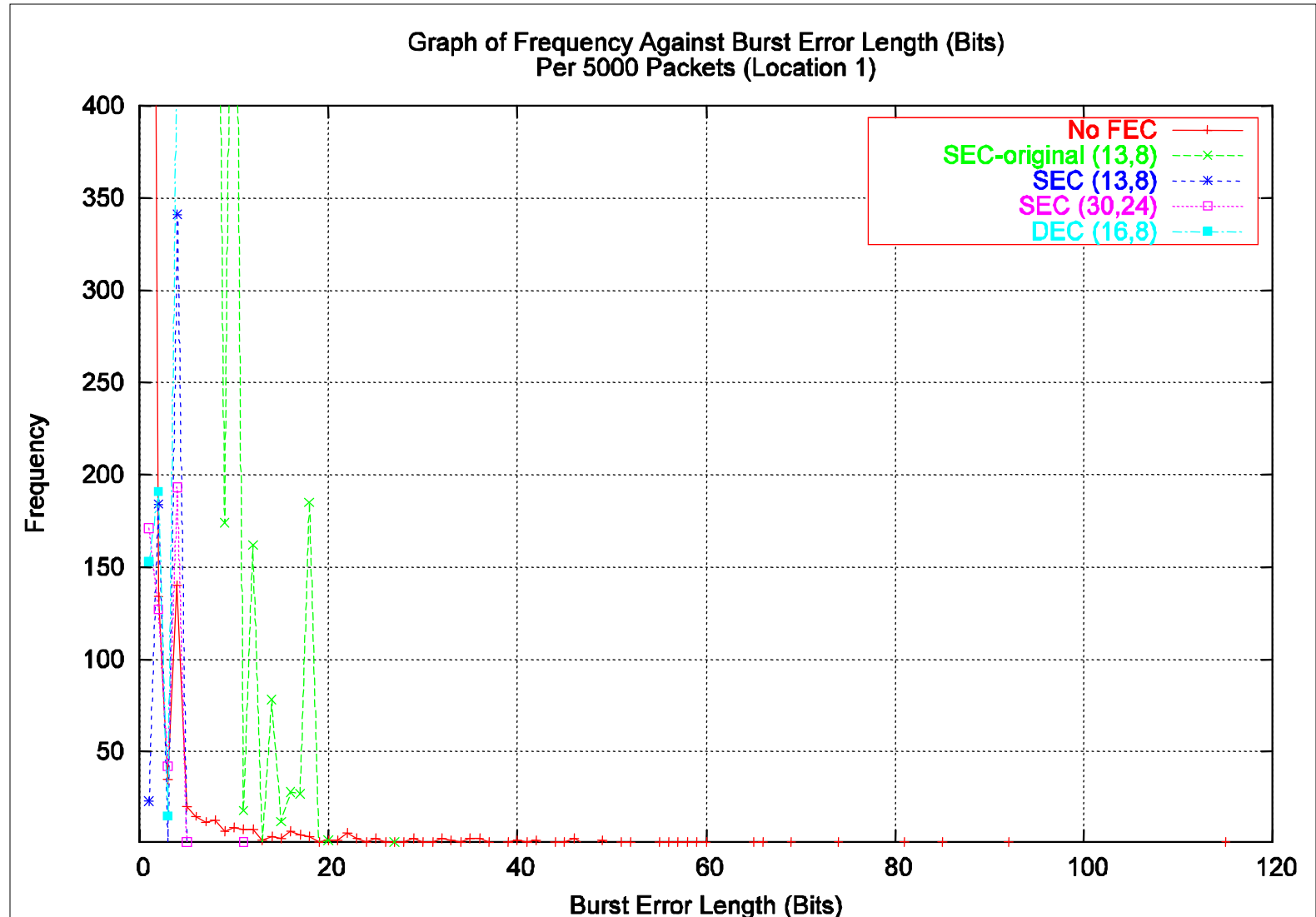
Results (Indoors)



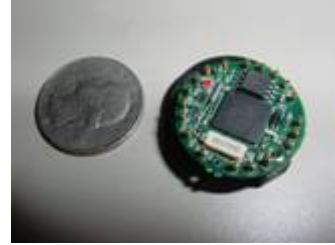
- We observed the following error patterns:
 - Mostly 1 or 2 bit errors, but non-negligible multiple bit errors
 - Small number of burst errors occurred and its occurrences depends on location.



Results (Burst bit errors)

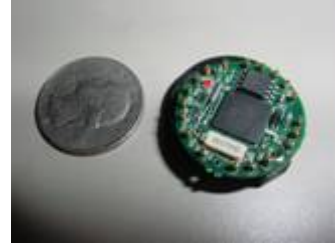


Conclusion



- We observed the following error patterns:
 - Most errors are 1 or 2 bit errors and they are well spread.
 - In indoor tests, the frequency multiple bit or burst errors increased.
- ECC was effective when error rate was low, but not so effective when error rate was high.
- SECDED 3 bytes performed well outdoors.
 - Low error rate
 - Faster than SECDEC 1 byte or DECTED.
- Using ECC was not effective when error rate is high (indoors)
 - Increased multiple bit or burst errors.
 - Needs CRC!

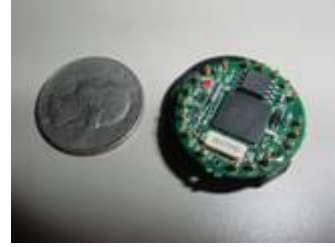
Question



Extra Slides



Glossary



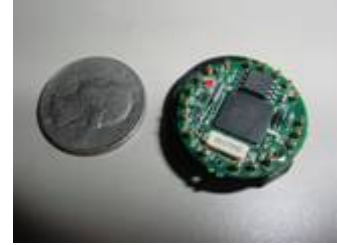
- **If a cyclic shift of $n \neq 0$ digits always produces another code word, then it is said to be quasi-cyclic.**



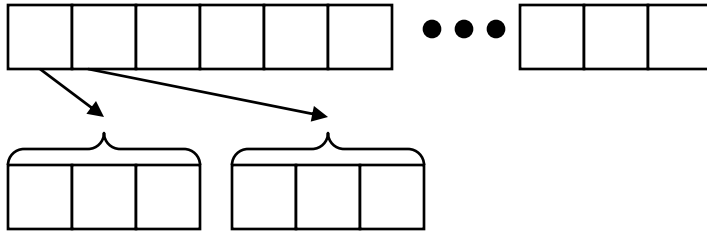
(16,8) Systematic QC Code

- Quasi-cyclic code can correct 2-bit errors.
- Cyclic structure
 - Encoding can be done serially using linear feedback shift register
 - Or in parallel
 - Possibly implemented in sensor mote hardware to reduce load on microcontroller
- For a (16,8) systematic quasi-cyclic code, can correct double errors in 16 bits
 - We do not differentiate between cases with 3 or more errors
 - Simply signal to higher layer that packet has errors
- Syndromes:
 - 137 possible 8-bit syndromes
 - 1 used by zero error (syndrome = 0)
 - 16 used by single correctable errors (encoded message is 16-bits, so ${}^{16}C_1$)
 - 120 used by double correctable errors (${}^{16}C_2$)

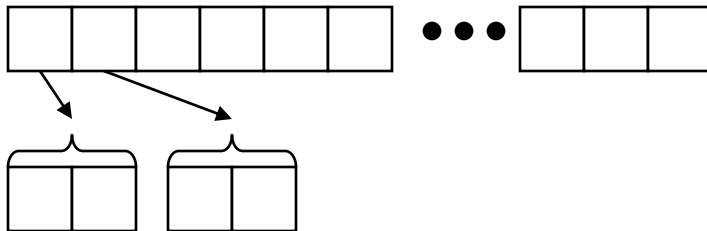
SECDED illustration



SEC Org



SEC 1 byte



SEC 3 bytes

