Selecting
data representation formats
for reduced energy

Does encoding matter?

Why optimize if the difference is small?

Data impacts on encoding size

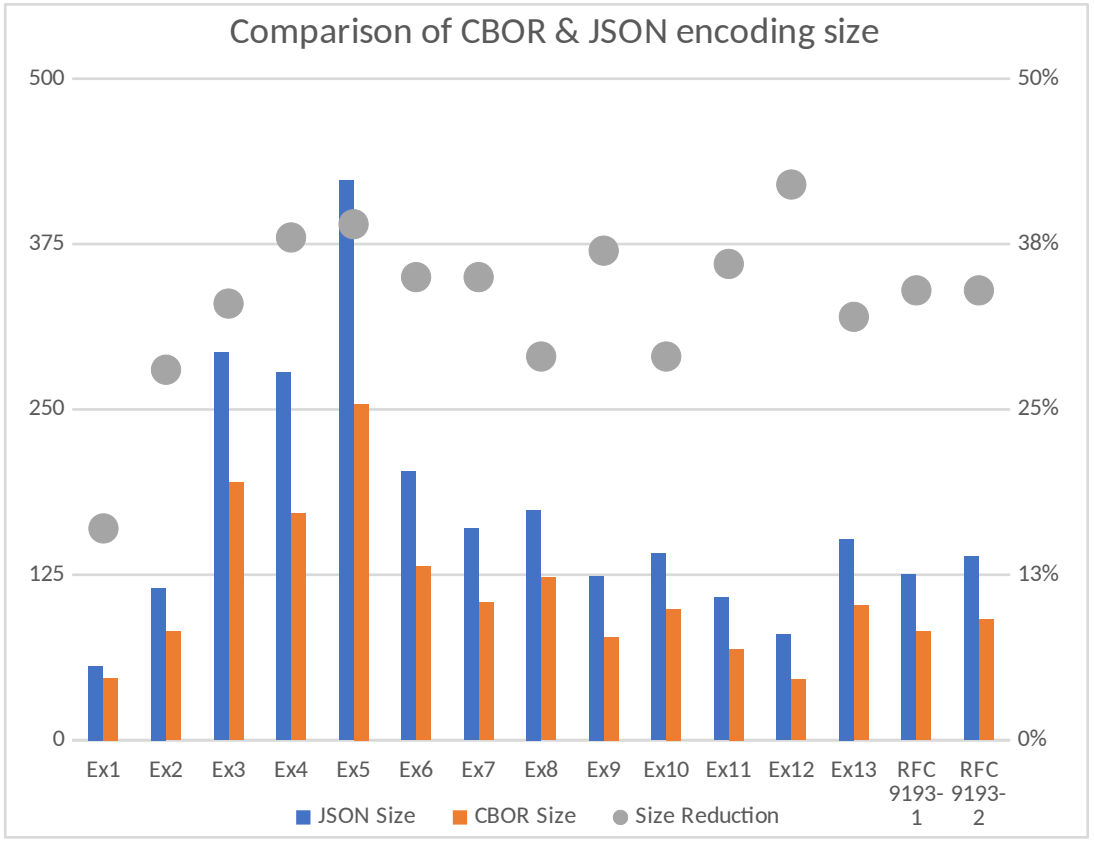
- Text data encodes into text formats well.
  - Non-text encodes poorly
  - Hex escape sequences produce 4x inflation of escaped octets
- Binary data encodes into text formats poorly:
  - Base64 = 33% data inflation
- Integers encode poorly into text
  - Typically 50% data inflation
- Floating point encodes poorly into text
  - Trivial examples are smaller than binary (e.g. 1.1)
  - Real examples are larger than binary (e.g. -1.01)
- Structures encode poorly into text
  - Separators, beginning and end markers are needed
  - Data inflation typically 2 + N – 1 for N elements (e.g. JSON)

Type	JSON Size	CBOR Size
string	strlen+2 + escaping	strlen + UINT(strlen)
octets (hex)	bytesize * 2	bytesize + UINT(bytesize)
octets (b64)	bytesize * 4/3	bytesize + UINT(bytesize)
int8	1 to 3	1 or 2
int16	1 to 5	3
int32	1 to 10	5
int64	1 to 19	9
float32	3 to 16	5
float64	3 to 23	9
Date	12	2 + UINT(days since 1970)

Practical differences in encoding data size

Size comparison of JSON vs CBOR in SenML Examples

- Data from SenML examples
- Encoded as both JSON and CBOR
- CBOR size reduction in all cases
  - Often 33% or better



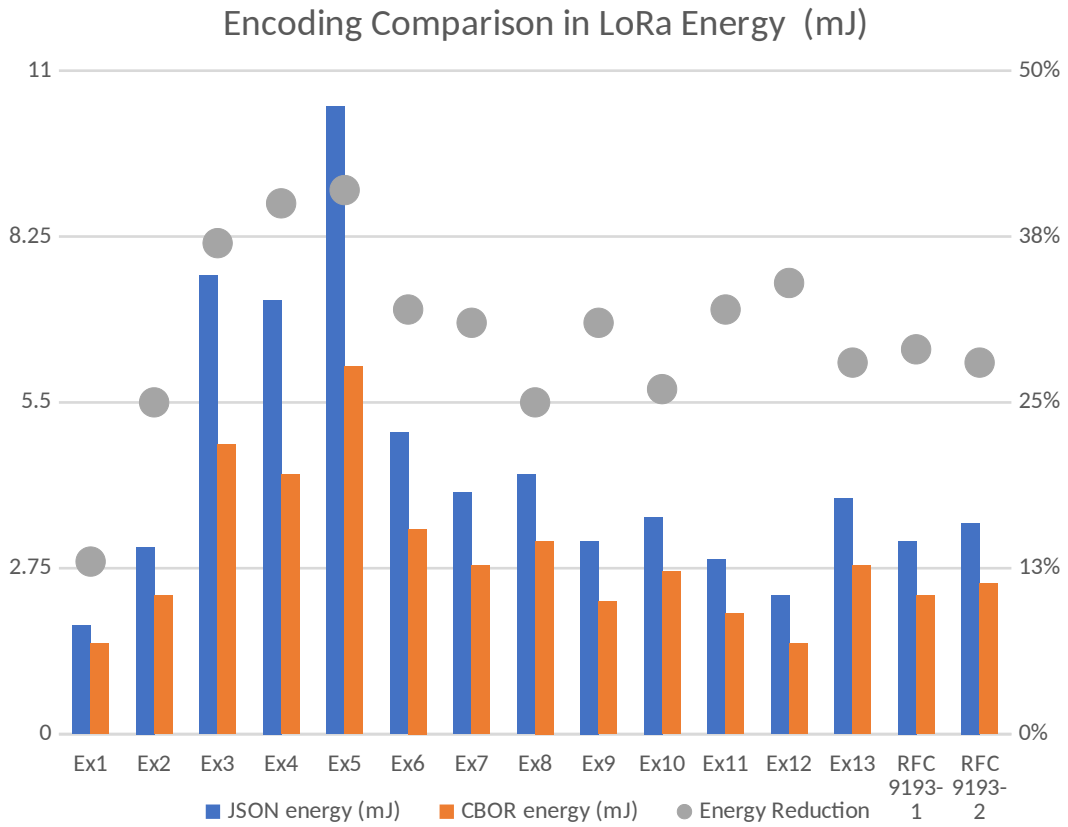
Encoding impact on energy

Why optimize if energy impact is small?

Energy impact of data size by encoding

Energy comparison of JSON vs CBOR in SenML Examples

- LoRa overhead reduces impact
  - Sensitive to packet count
  - Quantized to 127 bytes
  - Per-packet overhead
    - Favors reduction across packet count
- Energy reduction in all cases
  - Often 30% or better



Impact of energy reduction in constrained networks

- Smaller batteries
- Longer life
- Smaller need for energy harvesting
- Reduced e-waste (for primary cell)
- Lower cost

Encoding Choices in IETF

JSON & CBOR account for most hierarchical data formats

Common myths of text formats

Why people still think they like text formats

- “It’s easier to debug JSON”
  - Many tools for CBOR ➔ CBOR Debug
- “I don’t need to install a tool to look at JSON”
  - CBOR decoding can be done in a web browser

Unpleasant truth:

- These are tooling problems, not encoding problems.
- The vast majority of traffic is never debugged.
- Plan for primary use case: machine interpretation

Benefits of binary encodings

- Simple to parse
  - Low embodied energy
    - Low code
    - Low memory
  - Low active energy
    - Low compute overhead
- **Lower data use**
  - Lower transmit & receive energy
- Lower interpretation complexity
  - Simpler security posture
- **Less per-character work**
  - Escaping, delimiting
- **Less redundant conversion work**
  - Decimal conversion
  - Base64 encoding
- More deterministic
  - Whitespace
  - Escape choices

Recommendations

Suggestions to the IAB

- Consider content and intended use for data representation formats, e.g.:

Configuration documents	Text formats are appropriate
Primarily text content	Text formats are appropriate
Primarily non-text content	Binary formats should be preferred

- Not a game changer for e-impact, but a small contribution

Backup

Model use case assumptions

- Model use case: LoRa Class A node
- Justification:
  - Common LPWAN
  - Simple software stack
  - easy deployment
- Leaf Transceiver: SX1262
- Concentrator: SX1250 + SX1302
- LoRa packets are <=127 bytes
- Expect device energy to be dominated by radio energy
- See paper for network parameters