Implementation of BPv7-bis in µPCN

µPCN

- Implementation of DTN protocols for POSIX and ARM Cortex STM32F4 microcontrollers

Protocols

- Bundle Protocol v6 (RFC 5050)
- DTN IP Neighbor Discovery (IPND)
- Routing approach optimized for message-ferry micro LEO satellites

New protocol: BPv7-bis

- Integrate next revision of BP into µPCN
- Compare BPv7-bis implementation with existing BPv6 with respect to
    - *parsing time*
    - *bundle sizes*

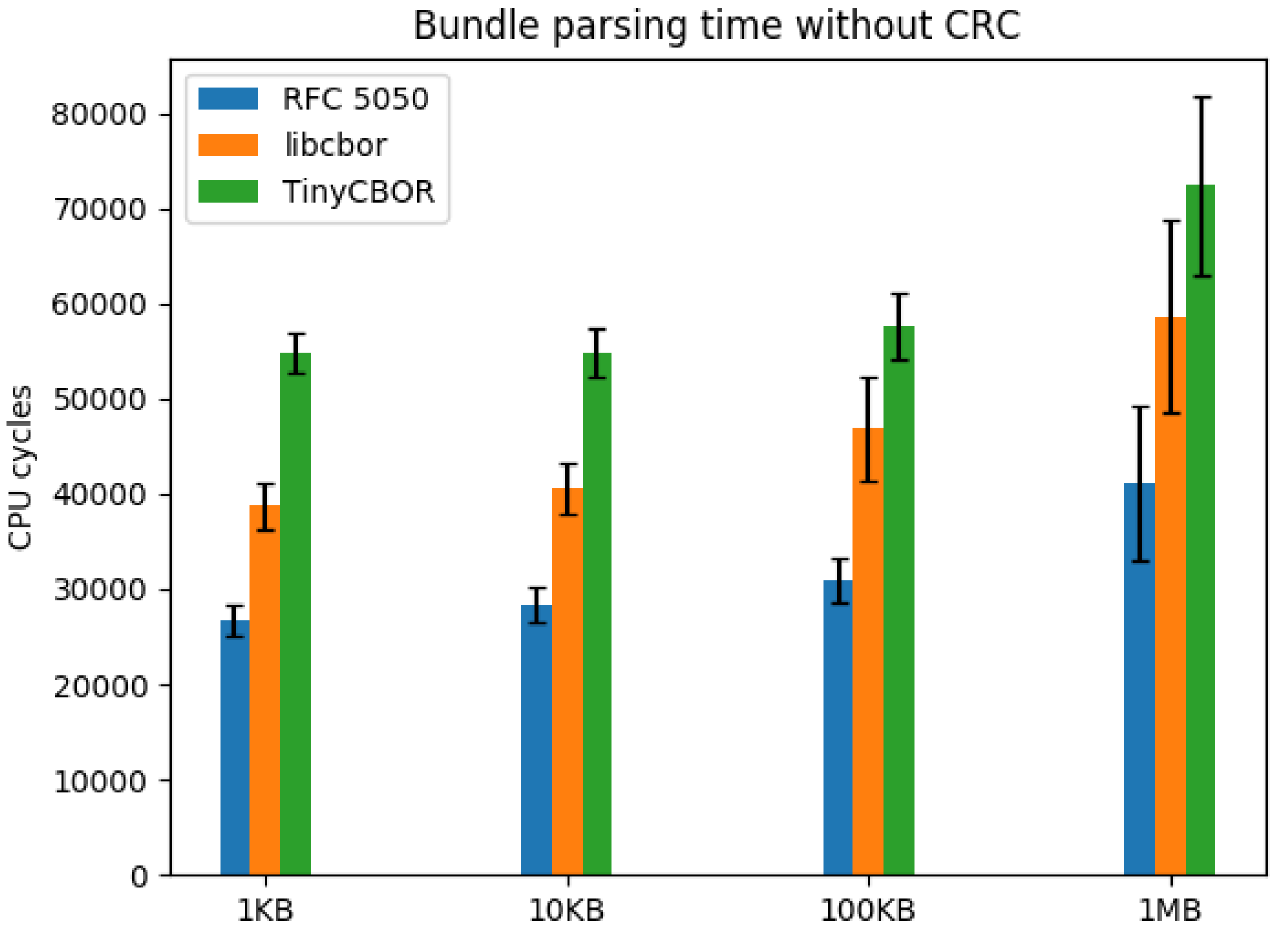
Comparison of bundle parsing times

Test Scenario

- Simple bundles with payload blocks of different size (1KB … 1MB)
- No other extension block than payload block
- Test different CBOR parsing libraries:
  - **libcbor**: incremental parsing with function pointer table
  - **TinyCBOR**: Iterator and if-else type checks
- CRC not used because it is not specified in RFC 5050 and therefore not comparable
- Binary bundles send to running µPCN instance

Performance Measurement

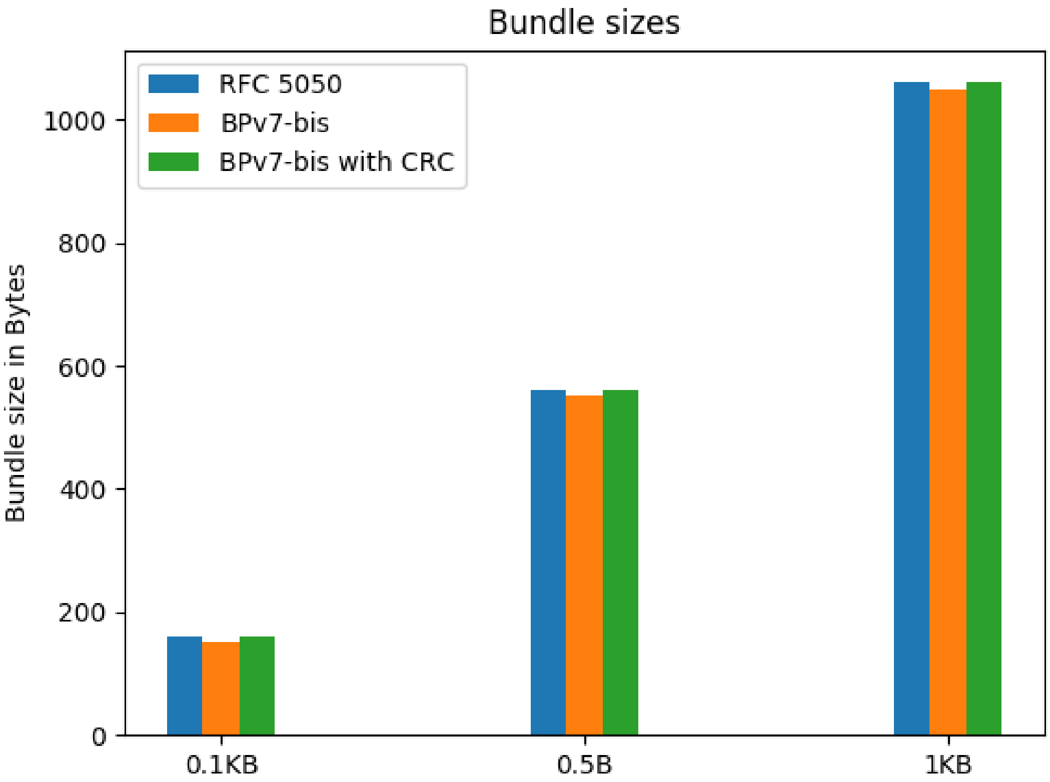
- Aggregation of hardware CPU cycles for each parsing step
- Measured with Linux `perf_event_open(2)`
  - `PERF_COUNT_HW_CPU_CYCLES` – total hardware CPU cycles



Results

- Significant overhead by CBOR (ca. 33% - 75%)
- Function pointers more efficient than explicit type checks for every field

Comparison of bundle sizes

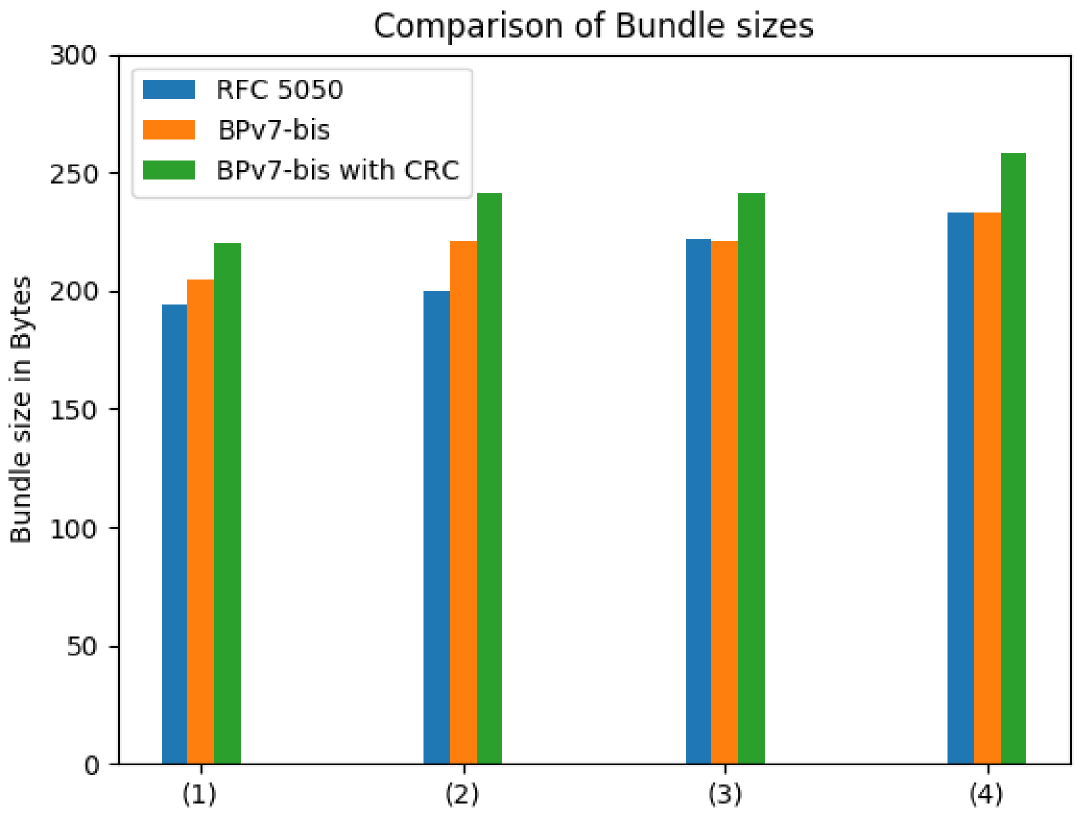


Test Scenario

- Simple Bundles with different payload blocks (0.1KB … 1KB)
- No other extension block than payload block
- BPv7-bis bundles generated with Python script, RFC 5050 bundles generated by µpcn

Result

- RFC 5050 bundles have nearly the same size as BPv7-bis bundles
- Smaller integer representation in CBOR is countered by extra structural CBOR information like array headers



Payload size: 128 Bytes

Comparsion of BPv7 EID encoding and RFC 5050 EID dictionary size
(1) Custodian = Report-To
(2) Custodian = Previous Node = Report-To
(3) Custodian != Previous Node != Report-To

Comparsion of multiple extension blocks blocks
(4) Hop Count + Bundle Age + Custom extension block with 19 bytes data (block was mimicked in RFC 5050 with SDNVs)