TECHNISCHE UNIVERSITÄT DRESDEN

DRESDEN concept

Marcel Beyer

Comparison of Bundle Protocol version 6 and 7

Dresden, August 1, 2019

Outline

Bundle Protocol

Changes v6-v7

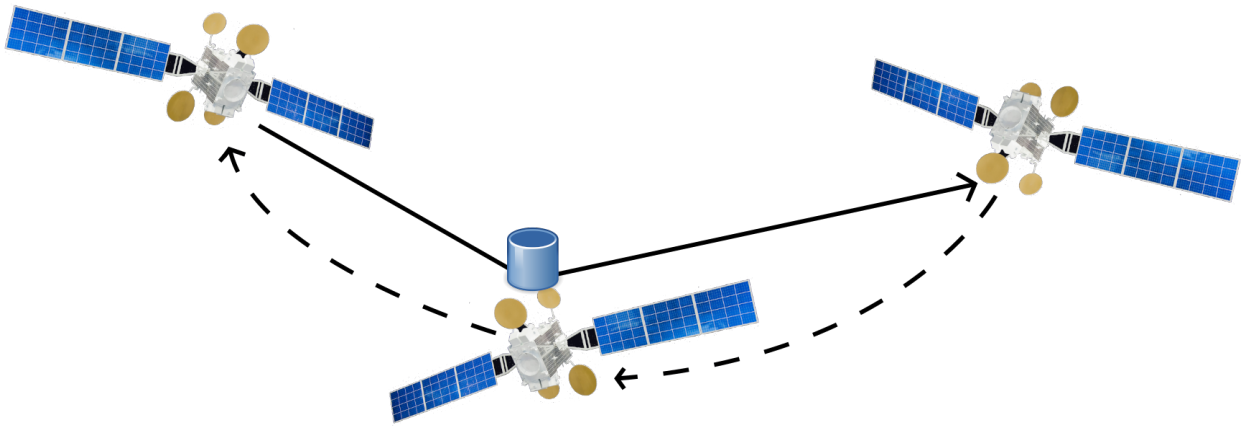
Demands

Implementations

Bundle Protocol I

- end-to-end protocol for communication in Delay Tolerant Networks (DTN)
- DTN are designed for sophisticated environments
  - intermittent connectivity
  - high/variable latency
  - high error rates
- Bundle Protocol (BP) sits on Application Layer of existing internets
  - Interface: Convergence Layer Adapter
- store-and-forward network
- Key capabilities:
  - use physical movement of data
  - ability to move responsibility for error control
  - cope with intermittent connectivity

Bundle Protocol II



- Bundle Nodes: Communicating entities
- Bundle: Metadata + Payload, data units sent over network
- no direct connection between sender/receiver
  - store-and-forward using other nodes

`https://de.wikipedia.org/wiki/Datei:AMOS-5_Satellite_--_with_star_background.jpg`
`https://de.wikipedia.org/wiki/Datei:Database.svg`

Unchanged fundamentals

- basic principles were kept
- components of nodes
  - Convergence Layer Adapters
  - Bundle Protocol Agent
  - Application Agent (Application Specific Element, Administrative Element)
- construction of bundles
- processes for sending, receiving, forwarding bundles
- fragmentation

Convergence Layer Adapter

Version 6	**Version 7**
many possible protocols, minimal set of needed services	
send bundle to *all* reachable nodes in minimal reception group	send bundle to *a* node that is reachable
forward received bundles to BPA	
list is neither exhaustive nor exclusive; supplementary DTN protocols may expect additional services	

TLV → CBOR

v6

- “classic” protocol: Type, Length, Value: fields in given order, known length
- SDNV for variable length of integers

v7

- SDNV came out as inconvenient
- CBOR: more datatypes, structured
- every block is encoded as CBOR array

SDNV: encode arbitary-length integers with minimal overhead
Example: $1_{10} = 00000001_2$; $128_{10} = 1000000100000000_2$

CBOR: binary data serialization format, inspired by JSON

Excluded Custody

Version 6:

- Custody Transfer ensures retransmit of lost packages
- nodes can accept custody for bundle
- node stores bundle as long as it has custody

Version 7:

- Custody moved from BP to new Bundle-in-Bundle-Encapsulation Protocol (BIBE)
- only used if necessary
- problems with custody: no partial retransmit, no NAKs

Custody/BIBE

Functional principle:

- node accepts custody for bundle
- custody is requested by bundle processing control flags
- node sends custody reports for bundles in custody
- bundles in custody are stored on node until another node has custody

- specified in v6 of BP
- specified as BIBE for v7 of BP

Node IDs

- Endpoint: 0 or more Nodes
- Need to identify a specific node for many purposes of BP

- v7 introduces “node IDs”
- each node has to be in a singleton endpoint
- endpoint id of this EP is node ID of the node
- node has to stay in this endpoint

- singleton endpoints already in v6
  - node was not forced to stay in this endpoint

Added CRC

- v6 without integrity checks
- in v7 all blocks have optional CRC
- 2 fields
  - CRC type
  - CRC value

Bundle Format

- $\geq$ 2 blocks
- first block: Primary Block with basic information to route bundles

- v6: “last block”-flag
- v7: Payload Block has to be last block

Restructured Primary Block

- changed to CBOR
- Primary Block is now immutable
- retained fields: version; bundle processing control flags; destination, source, report-to addresses, timestamp, lifetime, fragment offset
- removed dictionary for EIDs
  - address is now CBOR array with URI scheme-name and scheme specific part (SSP)
- removed fields for custody
- “block length”, “total application data unit length” removed
- added CRC type (and CRC)

Bundle Processing Control Flags

- order of bits changed
- retained fields:
  - bundle is fragment
  - bundle must not be fragmented
  - acknowledgment by application is requested
  - status report requests
- removed
  - custody (status) is requested
  - singleton destination
- added
  - status time is requested in status reports
  - bundle contains manifest block

Canonical Blocks

- CBOR
- all blocks except Primary Block
- retained fields
  - Block type code
  - Block Processing Control Flags
  - Block-type-specific data
- removed
  - Block data length
  - EID Reference Count and List
- added
  - Block number
  - CRC type, CRC

Block Processing Control Flags

- retained fields
  - send status report if node is unable to process block
  - delete block from bundle if processing impossible
  - delete bundle if processing impossible
  - replicate block in all fragments
- removed
  - last block
  - forwarded without processing
  - contains EID reference
- 4 new bits, reserved for future use

Extension Blocks

- Extension Blocks known in v6 and v7
- v6 has no concrete extension blocks specified
- v7 specifies 3 extension blocks, no exhaustive list
- nodes need to get along with unknown extension blocks

  v6: “Block was forwarded without being processed” flag
  v7: block processing control flags indicate action:
  - remove block from bundle
  - delete bundle
- v7 adds unique block IDs to extension blocks

Specified Extension Blocks in v7

Previous Node

- identifies the node that forwarded the bundle

Bundle Age

- time between bundle creation and last forwarding
- check lifetime expiration without accurate clock

Hop Count

- hop count and hop limit
- delete bundle when count exceeds limit
- removes bundles on forwarding errors

Summary

	V6	V7
Convergence Layer Adapter	send to all reachable nodes	send to a (single) node
Data format	bit pattern, SDNV	CBOR
Custody	Specified in BP	BIBE, Conv. Layer
Identifying single nodes	Singleton endpoints	constant Node IDs
Checksums	✗	every block
Bundle Format	“last block”-flag	Payload Block is last block
Primary Block	Dictionary for EIDs	Immutable
Canonical Blocks	References to EIDs	Block number
Extension Blocks	no concrete blocks specified	Previous Node, Bundle Age, Hop Count

Changed demands

- simplification
  - modularization (custody)
  - removing complex structures (dictionary)
- modernization, robustness (CBOR)
- future flexibility
  - CBOR
  - Extension Blocks

Implementations of v6/v7

implementation	version 6	version 7
DTN2	✓	✗
IBR-DTN	✓	proposed
ION	✓	anticipated
PyDTN	✗	✓
Terra	✗	✓
μPCN	✓	✓

K. Scott and S. Burleigh.
Bundle Protocol Specification.
RFC 5050, November 2007

S. Burleigh, K. Fall and E. Birrane.
Bundle Protocol Version 7.
Internet Draft, November 2018

C. Bormann and P. Hoffman.
Concise Binary Object Representation (CBOR).
RFC 7049, October 2013

W. Eddy and E. Davies.
Using Self-Delimiting Numeric Values in Protocols.
RFC 6256, May 2011

S. Burleigh.
Bundle-in-Bundle Encapsulation.
Internet Draft, January 2019