JOHNS HOPKINS
APPLIED PHYSICS LABORATORY

Status of WG and Personal Drafts

IETF 125 DTN WG

Brian Sipos
JHU/APL

Current WG Drafts

- BPSec COSE Context (draft-ietf-dtn-bpsec-cose-15)
- BP EID Patterns (draft-ietf-dtn-eid-pattern-05)
- UDPCLv2 (draft-ietf-dtn-udpcl-03)
- BP SAND (draft-ietf-dtn-bp-sand-02)

BPSec COSE Context

- Completed second WG last call in December
- Changes from -10 to -15:
  - Updated many descriptions in Section 2 for better explanations
  - Added explicit explanation of “AAD Scope special keys” with allowance for private/experimental use
  - Added requirement which restricts each security operation to a single COSE message result value
  - Separated Section 3.2 Interoperability Algorithms into different subsections per key family
  - Added “direct+HKDF” algorithm to interop. minimum table
  - Updated PKIX profile in Section 4 to be consistent with CA/Browser Forum profiles
  - Updated all examples to add BP-conformant CRC values and use example key strengths consistent with the CNSA 1.0 minimum
- Leaving out future capabilities enabled by COSE (especially PQC)
  - Later updates can easily raise the interop. minimum floor
  - Users are always free to use whatever COSE algorithms / parameters suit their needs
- COSE Context has completed interoperability testing in support of CCSDS use
  - Adds some more evidence to the maturity of this document

BP EID Patterns

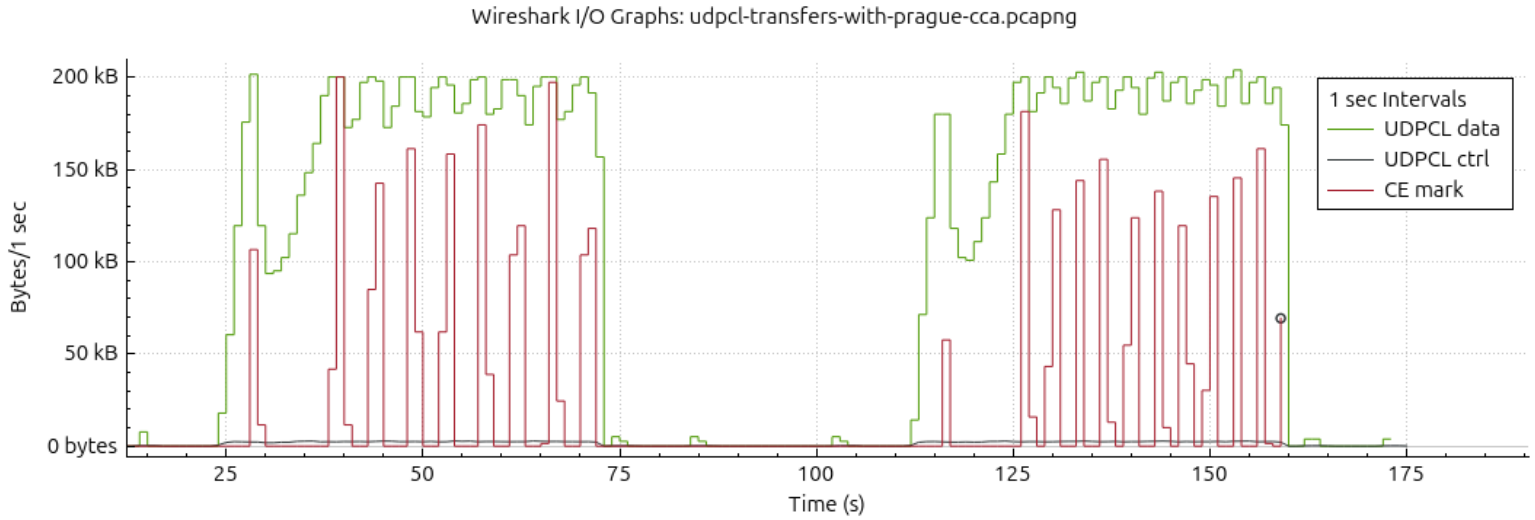
- Provides a framework for patterns on future EID schemes (*e.g.* IMC, IAC)
  - Adds a column to the IANA table to refer to the scheme-specific-pattern-defining specification
- Has multiple implementations from different orgs. in different languages
- Current variations for:
  - Match none (empty pattern)
  - Match all text form `*:**` exclusive of any other pattern items
  - Any-SSP text form `ipn:**|dtn:**` or `[ipn,dtn]:**`
  - IPN scheme-specific pattern, including wildcard and ranges of each element number
  - **No** DTN scheme-specific pattern specified
- Changes from -03 to -06:
  - Added more explicit terminology definitions
  - Allow the empty match-none pattern
  - More explicit Any-SSP logic and requirements (especially about normalization and elision)
  - More consistent IPN scheme pattern, handling 2-element text form only for single EID matching
  - Improved binary form range encoding using `ipn-range` rule and explanatory diagram
  - Added security considerations subsections
- Have identified a need for interoperability minimum support on:
  - Number of items in one pattern
  - Number of intervals in each IPN element range
- Would like a WG last call on the draft as-written or after this known issue is addressed

UDPCLv2

- Has had one trial implementation of all features
- Adds extensibility, segmentation, and transport security capabilities
- Includes compatibility mechanisms for:
  - In-band congestion control (IP ECN feedback)
  - Packetization-layer path MTU discovery (PLPMTUD)
  - Limited forms of other IP path characterization (reachability)
- Changes from -01 to -03:
  - Better explanation of optional-to-use behaviors
  - Added definition for “IP reachable”
  - Updated extension encodings using `int-range` generic CDDL rule
  - More specific requirements under Explicit Congestion Notification section 3.8
  - Added appendix B to explain endpoint congestion control algorithm (CCA) selection and use
- Remaining TBAs are for IP multicast assignments
  - Request to AD is pending!

UDPCL: IP Congestion Control Experiment

- Experiment using L4S (shallow) queuing and Prague CCA
- All IP logic implemented by COTS Linux kernel via POSIX socket API
  - Artificial throughput limit at 200 kB/s, with ECN congestion experienced (CE) marking before dropping
  - Could not easily simulate additional path effects such as loss, error, delay, or jitter with only two hosts
- Result scenario sends two 8 MiB bundles over UDPCL
  - Also includes some background BP and UDPCL control traffic
- Reaches near-steady rate close to the limit with steady average CE marking



BP Secure Advertisement and Neighborhood Discovery

- This draft is attempting to define what transport (CL) agnostic and baseline-secure discovery should look like
- Depends on:
  - UDPCLv2 for **zero-configuration** discovery
  - IMC multipoint scheme for **non-specific** advertisement
  - BP EID Patterns for **registered endpoint** advertisement
- Enables using any CL combination for **targeted** advertisement to known neighbors/peers
- Advertising capabilities:
  - Credentials (public key certificates)
  - Underlayer networks
  - CLA support
  - Resources (short-horizon power schedule)
  - Local topology (one-hop neighbors, routing metrics)
  - Routing willingness
  - Registered endpoints

BP SAND Draft Updates

- Updates from -01 to -02:
  - Added explicit terminology to align with IETF topology models
  - Allow multiple underlayer network (ULN) advertisement in one message
  - Use explicit termination point index to correlate between ULN, CL, and routing metrics advertisement
  - Added CL type code points for non-standard uses of LTPCL and “experimental UDPCL”
  - Removed uses of unpublished, alternative C509 PKI certificates
  - Fixed various inconsistencies or typos in CDDL rules
- Open discussion points:
  - Needs more implementation experience, specifically for fields needed for neighbor discovery
  - Depends on TBAs in the (not yet defined) IMC endpoint scheme

Current Personal Drafts

- BP Manifest Block (draft-sipos-dtn-manifest-block)
  - General purpose container for data derived from other, pre-existing blocks in the same bundle
  - This block type supports a variety of use cases explained in the draft
  - The manifest has been identified as a mechanism to support of SBAM
- BP Security Associations with Few Exchanges (SAFE) (draft-sipos-dtn-bp-safe)
  - This first individual draft provides a starting point for discussion
  - The initialization with EDHOC and primary SA agreement are in a good state
  - The ability to pre-position pre-keys and perform unilateral rekey is a unique feature
  - Need implementation experience for specific secondary SA use cases
- BP Edge Node with Zero-Configuration (draft-sipos-dtn-edge-zeroconf)
  - Goal is to allow using a BPA in a stable IP (v4 and/or v6) LAN without spending time configuring nodes
    - Allow BP application demos to focus on *using the apps* not on configuring the network
  - This is a purely informational document for how to use existing mDNS and DNS service discovery
  - Could be updated to include additional example of using existing UDPCL service name
  - A proof of concept was implemented in the dtn-demo-agent
  - Is there any interest from BPA implementers to demonstrate “plug and play” mDNS use?

Manifest Block Structure and Example Reason

- Each manifest block contains a mandatory “reason” code
- Each manifest block contains discretionary summary data:
  - Metadata of the manifest or bundle itself
  - Per-block derived data items chosen from a set of blocks present at the time of manifest creation
- A prime example of this use is a manifest containing Hash-of-BTSD for specific blocks
  - Allows the manifest to be of bounded, small size
  - Integrity can target the manifest
  - Verification of integrity is now decoupled from verification of the manifest (hashes)

```
; Manifest is encoded in BTSD content as a sequence
; EDITOR NOTE: block type 255 is a placeholder for IANA assignment
$extension-block /=
  extension-block-use<255, (bstr .cborseq ext-data-manifest / bstr)>
ext-data-manifest = [
  metadata-map,
  [* blockdata-map]
]

; General structure of all maps in the manifest
manifest-structure = {
  * int16 => any
}
int16 = -32768 .. 32767

; Metadata item socket
metadata-map = {* $$metadata-item } .within manifest-structure
; Block data item socket
blockdata-map = {* $$blockdata-item } .within manifest-structure
```

Figure 1: Manifest BTSD Content CDDL

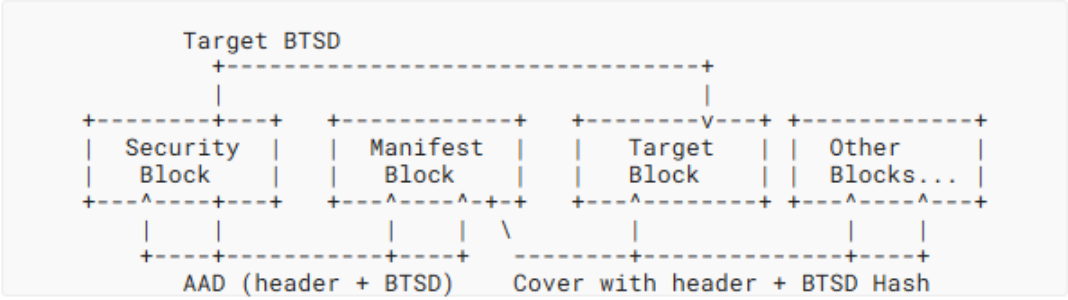


Figure 12: Security AAD Using a Manifest