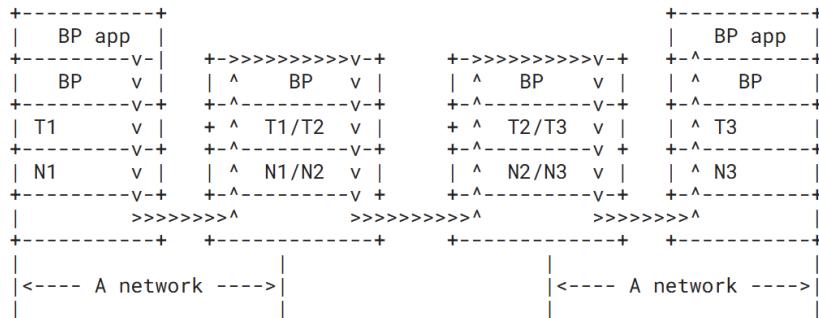
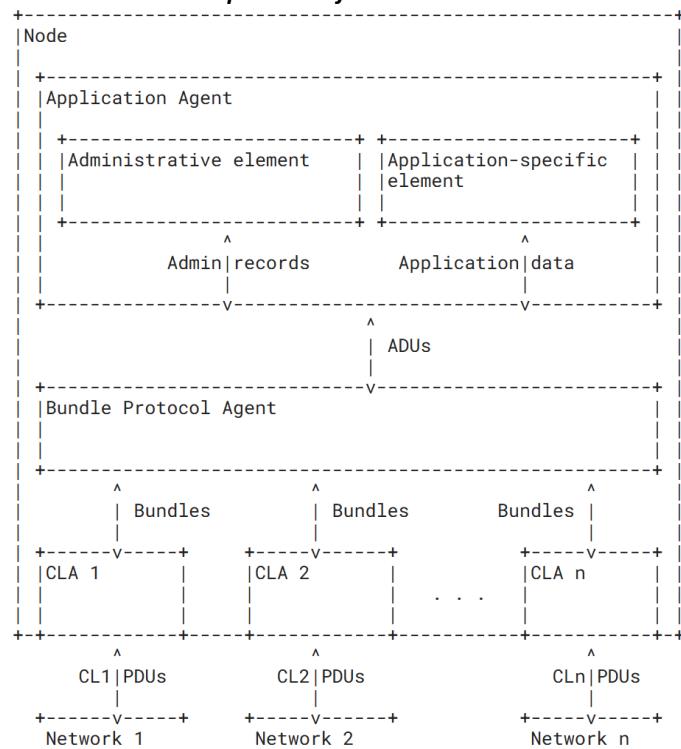


RFC 9171 Bundle Protocol v7

The Bundle Protocol in the Protocol Stack Model



Components of a Bundle Node



WIRESHARK TIPS FOR ION ANALYSIS

Download custom Wireshark profiles and trace files from:



- Use most recent version of Wireshark for best dissectors for BP. Download from <https://wireshark.org>
- You can apply filters based on IP address of nodes to find the ION traffic (if LTP, TCPCL, and Bundle dissectors are not visible).
- When using non-standard ports, right-click on packet in Packet List Pane and choose Decode As | select port number | select TCPCL or LTP in the Current column.
- Check out the dissectors at <https://gitlab.com/wireshark/wireshark>:
 - `epan/dissectors/packet-bpv7.c`
 - `epan/dissectors/packet-ltp.c`
 - `epan/dissectors/packet-tcpcl.c`

DISPLAY FILTERS – ION TRAFFIC	NOTES
<code>ltp</code>	Contains Linklayer Transmission Protocol
<code>tcpcl</code>	Contains TCP Convergence Layer
<code>bpv7</code>	Contains Bundle protocol
<code>tcpcl.data.proc.start == True</code>	Segment contains start of bundle
<code>tcpcl.data.proc.end == True</code>	Segment contains end of bundle
<code>tcpcl.pkt_type == 4</code>	TCPCL Keepalive
<code>tcpcl.contact_hdr.magic == 64:74:6e:21</code>	TCPCL Contact Header
<code>ltp.block.red_size > 0</code>	LTP red
<code>ltp.block.green_size > 0</code>	LTP green
<code>ltp.cancel.code == 0x02</code>	LTP Cancel
<code>bpv7.primary.bundle_flags.user_app_ack == 1</code>	Bundle – App ACK requested
<code>bpv7.primary.bundle_flags.no_fragment == 1</code>	Bundle – No fragment is set
<code>bpv7.primary.lifetime < 100000</code>	Bundle – Lifetime < 100 seconds
<code>bpv7.canonical.type_code == 1</code>	Bundle with Bundle Payload block
<code>bpv7.canonical.type_code == 6</code>	Bundle with Previous Node block
<code>bpv7.canonical.type_code == 7</code>	Bundle contains Bundle Age block
<code>bpv7.canonical.type_code == 10</code>	Bundle contains Hop Count block
<code>frame contains "bping"</code>	Detect bping (default payload)

CHAPPELL UNIVERSITY

Compliments of Chappell University (<https://chappell-university.com>)

Acronym	Definition
ADU	Application Data Unit (payload from the app to BP)
ARQ	Automatic Repeat reQuest
BAB*	Bundle Age Block (BPv7) - Bundle Authentication Block in BPv6
BCB	Block Confidentiality Block (encryption/confidentiality)
BDT	Best Delivery Time (routing)
BDT	Best-case Delivery Time (routing)
BFS	Breadth First Search (algorithm for route discovery)
BIB	Block Integrity Block (integrity/authentication)
BIBE	Bundle-in-Bundle Encapsulation (tunneling)
BP	Bundle Protocol (DTN network layer)
BPA	Bundle Protocol Agent (the node's BP engine)
bpadmin	Bundle Protocol configuration (endpoints, schemes, ducts)
bpchat	Send input text in bundles
bpclock	BP timer/housekeeping daemon
bping	Ping over BP (similar to, but not ICMP)
bprecvfile	Receive files
BPSec	Bundle Protocol Security framework for BPv7
bpsink	Receive bundles and print contents
bpsource	Send bundles via stdin/stdout
bptrace	Send a trace payload and report the path
bptransit	Bundle forwarding/transit daemon
BPv7	Bundle Protocol Version 7
CBHE	Compressed Bundle Header Encoding
CBOR	Concise Binary Object Representation (binary serialization used by BPv7/BPSec)
CCSDS	Consultative Committee for Space Data Systems
CFDP	CCSDS File Delivery Protocol
cfdpadmin	CFDP configuration
CG	Contact Graph (time-ordered contacts/ranges used by CGR)
CGR	Contact Graph Routing (time-dependent routing over scheduled contacts)
CID	LTP Session ID (identifies an LTP session)
CL	Convergence Layer (adapts BP to a transport)
CLA	Convergence-Layer Adapter/Agent (driver/implementation)
COSE	CBOR Object Signing and Encryption (used by many BPSec ciphersuites)
COTS	Commercial off-the Shelf
CP	Checkpoint (marks reliable red-part data for reporting - LTP)
CP	Contact Plan (routing)
CRC	Cyclic Redundancy Check
CRP	Contact Review Procedure
CS	Ciphersuite (algorithm/profile identifier)
CSID	Client Service ID (maps LTP to upper-layer service like BP)
CSP	Contact Selection Procedure
DGR	Datagram Retransmission (lightweight UDP-based CL library)
DS	Data Segment
DSN	Deep Space Network
DTKA	Delay-Tolerant Key Administration (DTN/BPsec key management)
DTLSR	Delay-Tolerant Link-State Routing
DTN	Delay/Disruption-Tolerant Networking
DTN Time	Seconds since the DTN epoch (2000-01-01 00:00:00)
EAT	Earliest Arrival Time (CGR routing metric)
EID	Endpoint Identifier (BP addressing)
EOF	End-Of-File (CFDP control PDU)

Acronym	Definition
ETO	Earliest Transmission Opportunity (routing)
EVC	Estimated Volume Consumption (routing)
EVL	Effective Volume Limit (routing)
FBTX	First Byte Transmission time (routing)
FDU	File Delivery Unit (CFDP data unit)
FRAG	Fragmentation (partial bundles for intermittent links)
GEO	Geostationary Earth-Orbit
GSL	Ground-to-Space Links
HCB	Hop Count Block (limits number of hops)
Inter-RR	Inter Region Routing
ionadmin	ION core configuration tool (contacts, ranges, SDR, etc.)
ionconfig	ION node configuration parameters file (for ionadmin)
ionlog	utility to redirect stdn to the ION log file
ionrc	ION node management commands file (for ionadmin)
ionsecadmin	ION security configuration
ionstart	Startup wrapper script for ION components
IPN	InterPlanetary Network (BP URI scheme like ipn:node.service)
ipnadminep	IPN administrative endpoint agent
ipnfw	IPN-scheme forwarder
IRTF	Internet Research Task Force
ISL	Inter-Satellite Links
LBRX	Last Byte Reception time (routing)
LBTX	Last Byte Transmission time (routing)
LEO	Low-Earth Orbit
LTP	Licklider Transmission Protocol (DT-tolerant transport with ARQ)
ltpadmin	LTP configuration (inducts/outducts)
ltpcli/ltpclo	LTP induct/outduct drivers
MAV	Maximum Available Volume (routing)
MEO	Medium-Earth Orbit
MIB	Management Information Base (CFDP configuration)
NAK	Negative Acknowledgment
OBC	On-Board Computer
OCGR	Opportunistic CGR
OWLT	One-Way Light Time (propagation delay)
PB	Primary Block (mandatory header block)
PDU	Protocol Data Unit (CFDP packet)
POSIX	Portable Operating System Interface
QoS	Quality of Service / Priority (bulk/normal/expedited in BP)
RA	Report Acknowledgment
RS	Report Segment (receiver status report)
SABR	Schedule Aware Bundle Routing
SDNV	Self-Delimiting Numeric Value (compact integer encoding)
SDR	Simple Data Recorder (ION's persistent shared-memory store)
TCPCL	TCP Convergence-Layer (BP over TCP)
tcpcli/tcpclo	TCPCL induct/outduct drivers
TLV	Type-Length-Value (CFDP metadata format)
TTL	Time To Live (bundle lifetime)
UDPCL	UDP Convergence-Layer (BP over UDP)
USLP	Unified Space Data Link Protocol
UTCG	Gregorian Coordinated Universal Time
ZCO	Zero-Copy Object (ION payload/buffer mechanism)

ION WATCH CHARACTERS

<https://ion-dtn.readthedocs.io/en/ion-open-source-4.1.4-a.1/ION-Watch-Characters/>

Bundle Protocol Watch Characters -----

a` - new bundle is queued for forwarding; `(nnn,sss,ttt,ccc)a`
`b` - bundle is queued for transmission; `(nnn,sss,ccc)b`
`c` - bundle is popped from its transmission queue; `(nnn,sss,ccc)c`
`m` - custody acceptance signal is received
`w` - custody of bundle is accepted
`x` - custody of bundle is refused
`y` - bundle is accepted upon arrival; `(nnn,sss,ttt,ccc)y`
`z` - bundle is queued for delivery to an application; `(nnn,sss,ttt,ccc)z`
`~` - bundle is abandoned (discarded) on attempt to forward it; `(nnn,sss,ccc)`~
`!` - bundle is destroyed due to TTL expiration; `(nnn,sss,ccc)!`
`&` - custody refusal signal is received
`#` - bundle queued for re-forwarding due to CL protocol failure; `(nnn,sss,ccc)`#
`j` - bundle placed in "limbo" for possible future re-forwarding; `(nnn,sss,ccc)`j
`k` - bundle removed from "limbo" and queued for re-forwarding; `(nnn,sss,ccc)`k

Enhanced watch characters prepends additional state information to the standard watch characters inside a pair of parenthesis. In this document, we use the following notion regarding enhanced watch characters information.

nnn = source node number
sss = service number
ttt = bundle creation time in milliseconds Epoch(2000)
ccc = bundle sequence number
xxx (LTP session number)

LTP Watch Characters -----

`d` - bundle appended to block for next session
`e` - segment of block is queued for transmission
`f` - block has been fully segmented for transmission; `(xxxx)f`
`g` - segment popped from transmission queue;
- `(cpxxx)g` -- checkpoint, could be a data segment or a standalone check point
- `(dsxxx)g` -- non-check point data segment
- `(rcpxxx)g` -- retransmitted checkpoint
- `(prsxxx)g` -- positive report (all segments received)
- `(nrssxx)g` -- negative report (gaps)
- `(rrssxx)g` -- retransmitted report (either positive or negative)
- `(rasxxx)g` -- a report ack segment
- `(csxxx)g` -- cancellation by block source
- `(crxxx)g` -- cancellation by block receiver
- `(caxxx)g` -- cancellation ack for either CS or CR

LTP Watch Characters (continued) -----

'h` - positive ACK received for block, session ended; `(xxx)h`
's` - segment received
- `(dxxx)s` -- received data segement with session number xxx
- `(rsxxx)s` -- received report segement with session number xxx
- `(rasxxx)s` -- received report ack segement with session number xxx
- `(csxxx)s` -- received cancel by source segement with session number xxx
- `(cas<xx)s` -- received cancel by source ack segement with session number xxx
- `(crxxx)s` -- received cancel by receiver segement with session number xxx
- `(carxxx)s` -- received cancel by receiver ack segement with session number xxx
't` - block has been fully received
'@` - negative ACK received for block, segments retransmitted; `(xxx)@`
'=' - unacknowledged checkpoint was retransmitted; `(xxx)=`
'+' - unacknowledged report segment was retransmitted; `(xxx)+`
'{` - export session canceled locally (by sender)
'}` - import session canceled by remote sender
'[` - import session canceled locally (by receiver)
']` - export session canceled by remote receiver

BIBECT Watch Characters -----

'w` - custody request is accepted (by receiving entity)
`m` - custody acceptance signal is received (by requester)
`x` - custody of bundle has been refused
`&` - custody refusal signal is received (by requester)
`\$` - bundle retransmitted due to expiration of custody request timer

BSSP Watch Characters -----

'D` - bssp send completed
'E` - bssp block constructed for issuance
'F` - bssp block issued
'G` - bssp block popped from best-efforts transmission queue
'H` - positive ACK received for bssp block, session ended
'S` - bssp block received
'T` - bssp block popped from reliable transmission queue
'-' - unacknowledged best-efforts block requeued for reliable transmission
'*` - session canceled locally by sender

<https://ion-dtn.readthedocs.io/en/ion-open-source-4.1.4-a.1/ION-Watch-Characters/>

TCPDUMP TIPS

<https://www.tcpdump.org/manpages/tcpdump.1.html>

Installation and Setup -----

```
sudo apt install tcpdump  
cd ~  
mkdir traces && cd traces
```

Key tcpdump Options -----

-c count

Exit after receiving or reading count packets.

-D

Print the list of the network interfaces available. Use the interface number with the -i flag to specify an interface on which to capture.

-h

Print the tcpdump and libpcap version strings, print a usage message, and exit.

-i interface number

On all supported Linux systems, as well as on recent versions of macOS and Solaris, an interface argument of ``any'' means a special pseudo-interface, which captures packets from all regular network interfaces of the OS (MAC header will appear as "Linux Cooked Capture".) If unspecified, tcpdump searches the system interface list for the lowest numbered, configured up interface (excluding loopback), which may turn out to be, for example, "eth0".

-n

Don't convert addresses (i.e., host addresses, port numbers, etc.) to names.

-# (enter a number)

Print a packet number at the beginning of the line.

-Q direction

Choose send/receive direction direction for which packets should be captured.

Possible values are 'in', 'out' and 'inout'. Not available on all platforms.

-q

Quick output. Print less protocol information so output lines are shorter.

-r file

Read packets from file (which was created with the -w option or by other tools that write pcap or pcapng files).

-s snaplen

Grab snaplen bytes of data from packets rather than the default of 262144 bytes.

Key tcpdump Options (continued) -----

-t

Don't print a timestamp on each dump line.

-ttt

Print a delta (microsecond or nanosecond resolution depending on the --time-stamp-precision option) between current and previous line on each dump line. The default is microsecond resolution.

-U

Write to the output right away, rather than being written only when the output buffer fills.

-v (also -vv and -vvv)

When parsing and printing, produce (three variations) verbose output.

-w file

Write the raw packets to file rather than parsing and printing them out.

expression

Apply pcap filter to select which packets will be dumped.

For the filter expression syntax, see

<https://www.tcpdump.org/manpages/pcap-filter.7.html>

Sample Capture Setups -----

Remove need for sudo (optional) – may persist through reboots

```
sudo setcap 'cap_net_raw,cap_net_admin=eip' "$(command -v tcpdump)"
```

Otherwise, preface each command with sudo

Capture traffic to or from 10.0.0.113 (on interface #1)

```
tcpdump -i1 host 10.0.0.113
```

Capture traffic from 10.0.0.113 (on any interface)

```
tcpdump -iany ip src 10.0.0.113
```

Capture traffic and show count. Write to a file called test1.pcap (any interface)

```
tcpdump -iany -v -U -w bpingtest_node11.pcap
```

Capture traffic to any 224.0.0.x destination IP addresses (any interface)

```
tcpdump -iany dst net 224.0.0.0/24
```

Capture traffic to or from TCP port 5446 (on interface #2)

```
tcpdump -i2 -v -U tcp port 5446
```