

103rd IETF/IRTF, Nov. 2018, Bangkok

# CCNinfo: Discovering Content and Network Information in Content-Centric Networks

---

draft-irtf-icnrg-ccninfo-00

Hitoshi Asaeda (NICT)

Xun Shao (KIT)

# History

---

- Initial proposal: Contrace
  - “Contrace: Traceroute Facility for Content-Centric Network”
    - IEEE ComMag, Mar. 2015 (no TLV format, external daemon process interacting with ccnd)
    - <https://tools.ietf.org/html/draft-asaeda-icnrg-contrace-00>, Oct. 2016 (CCNx TLV format)
- Revised with CCNinfo
  - “CCNinfo: Collecting Content and Network Information in Content-Centric Networks”
    - <https://tools.ietf.org/html/draft-asaeda-icnrg-ccninfo-00>, Mar 2018
    - <https://tools.ietf.org/html/draft-irtf-icnrg-ccninfo-00>, Oct. 2018

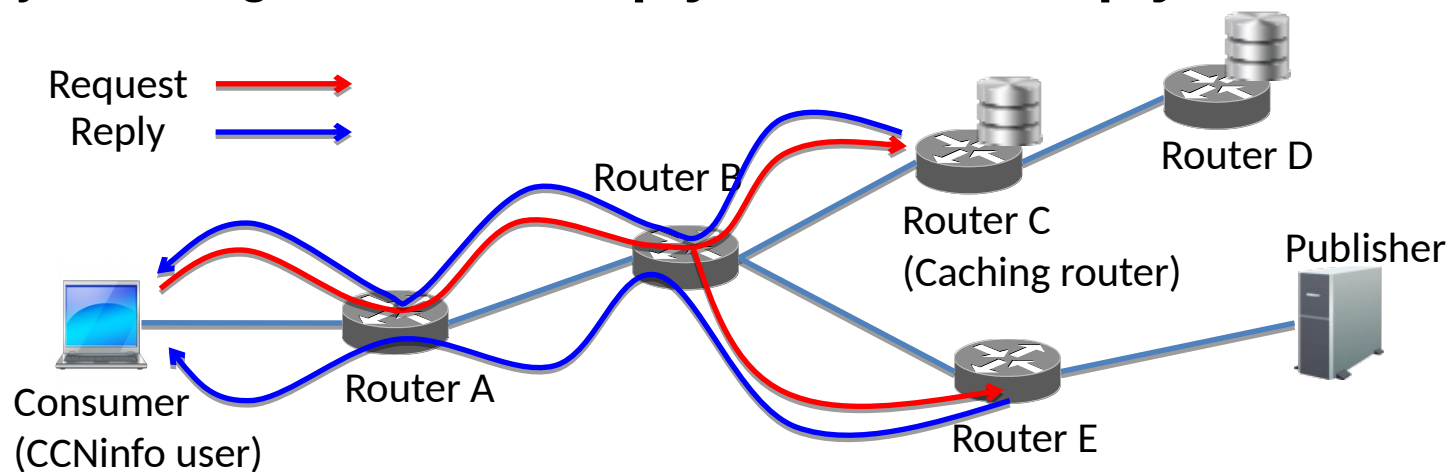
# What CCNinfo Can Provide

---

- Reachability of caching routers and publishers
- Hop count and RTT for content retrieval
- Path stretch
- Multipath condition
- Content popularity or cache hit ratio
- Chunk lifetime or expiration time
- CS usage at router
- Num. of interests per content
- (Access control and policy configuration for information disclosure)

# CCNInfo Basic Behavior

- **Request message** is initiated by CCNInfo user and forwarded toward caching router or publisher based on the FIB in a hop-by-hop manner
- Request message includes **Request block** and **Report block(s)**
- **Reply message** is initiated by caching router or the first-hop router (or publisher) and forwarded back toward CCNInfo user along the PIT entry
- Reply message includes **Reply block** and **Reply sub-block(s)**

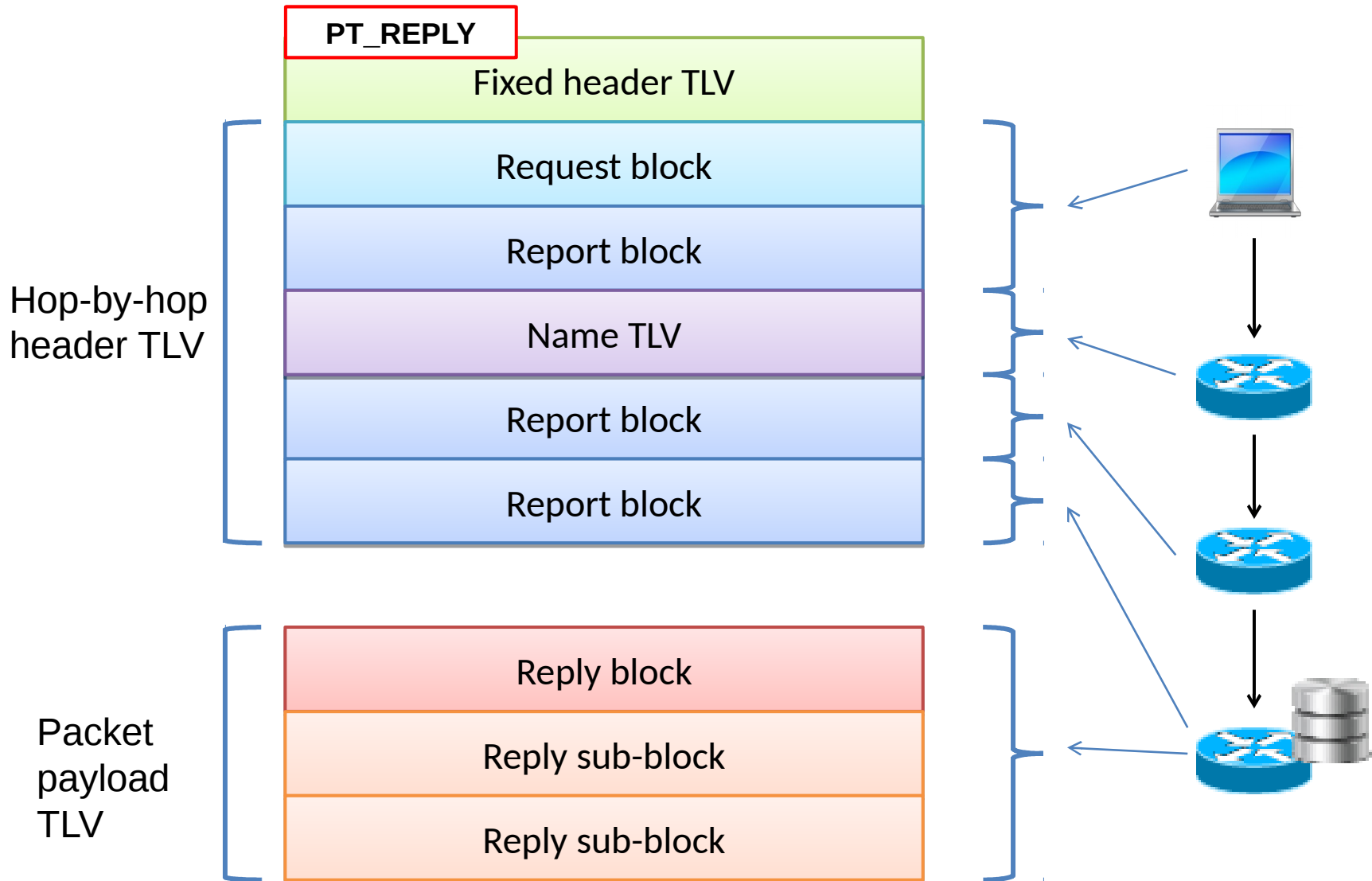


# CCNinfo Request/Reply Messages

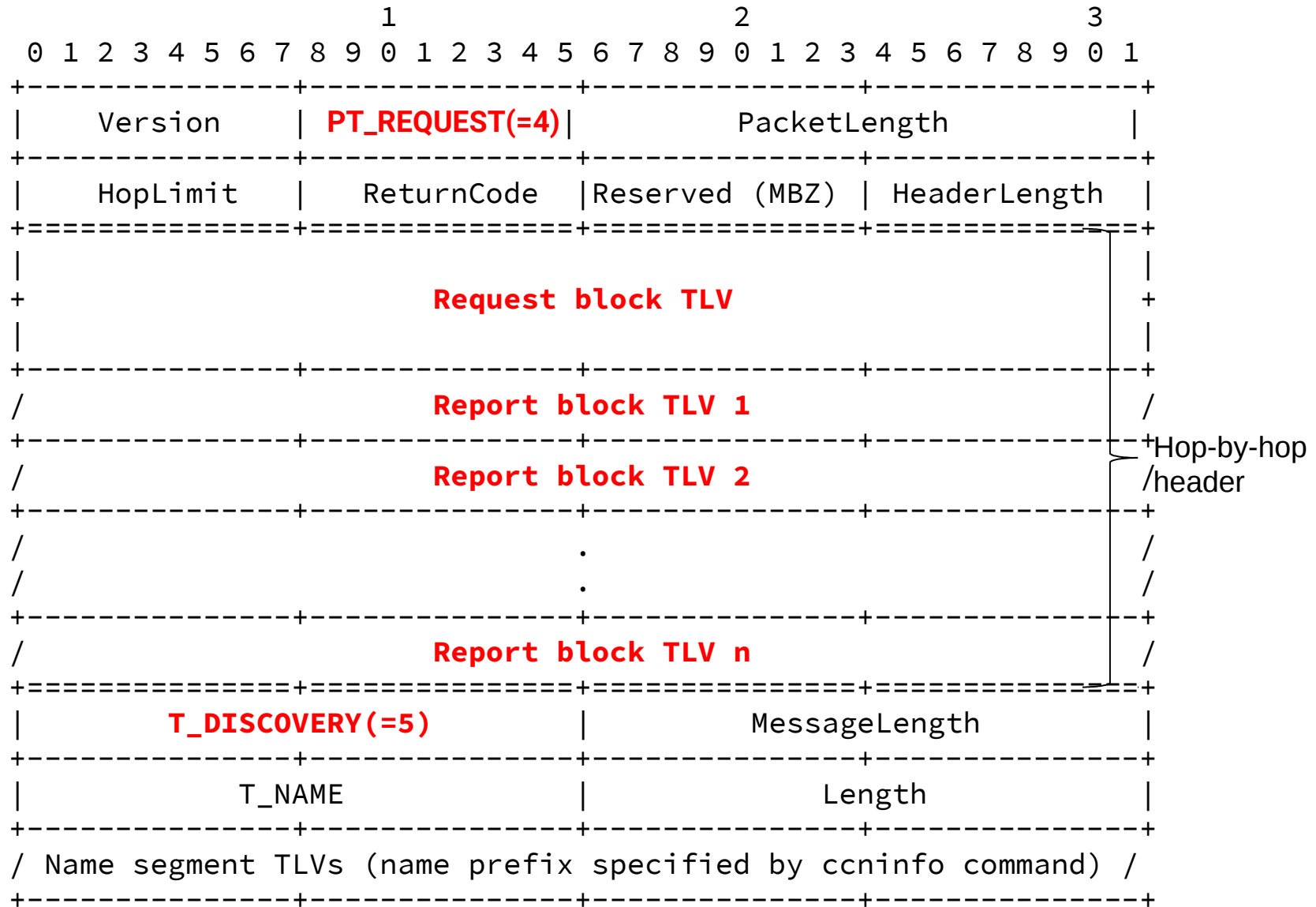
---

- Compatible with CCNx-1.0 TLV format
- CCNinfo Request Message
  - Request message consists of a fixed header, Request block TLV, Report block TLV(s), and Name TLV
- CCNinfo Reply Message
  - Reply message consists of a fixed header, Request block TLV, Report block TLV(s), Name TLV, and Reply block/sub-block TLV(s)
- New type values defined
  - Packet type; Request (PT\_REQUEST) and Reply (PT\_REPLY)
  - Top level type (T\_DISCOVERY)
  - Hop-by-hop type (T\_DISC\_REQ and T\_DISC\_REPORT)
  - CCNx message type (T\_DISC\_REPLY)

# CCNinfo Message Flow

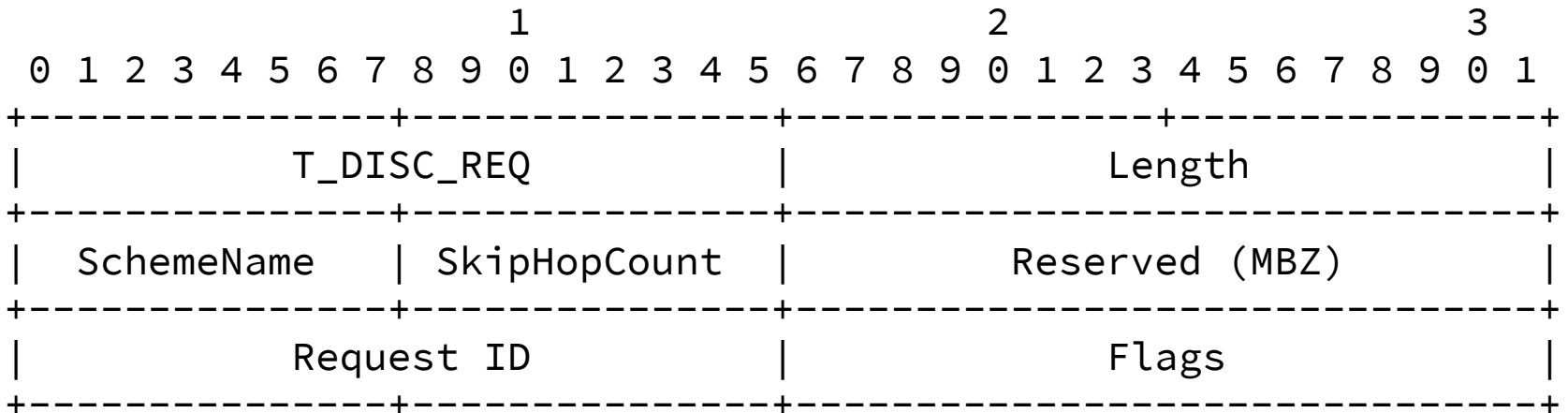


# Request Message

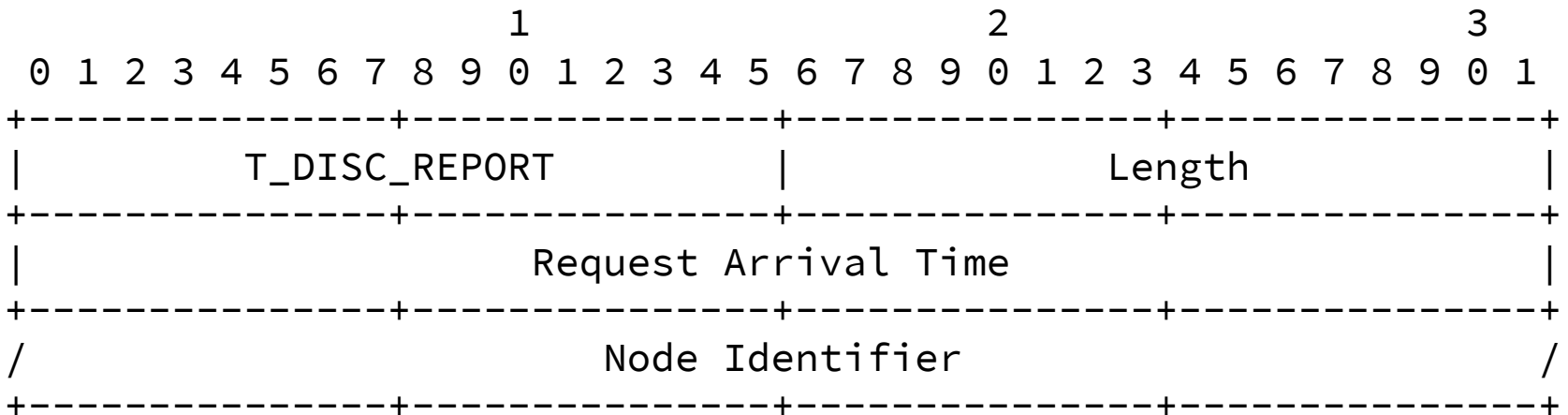


# Request Block and Report Block

## Request block TLV

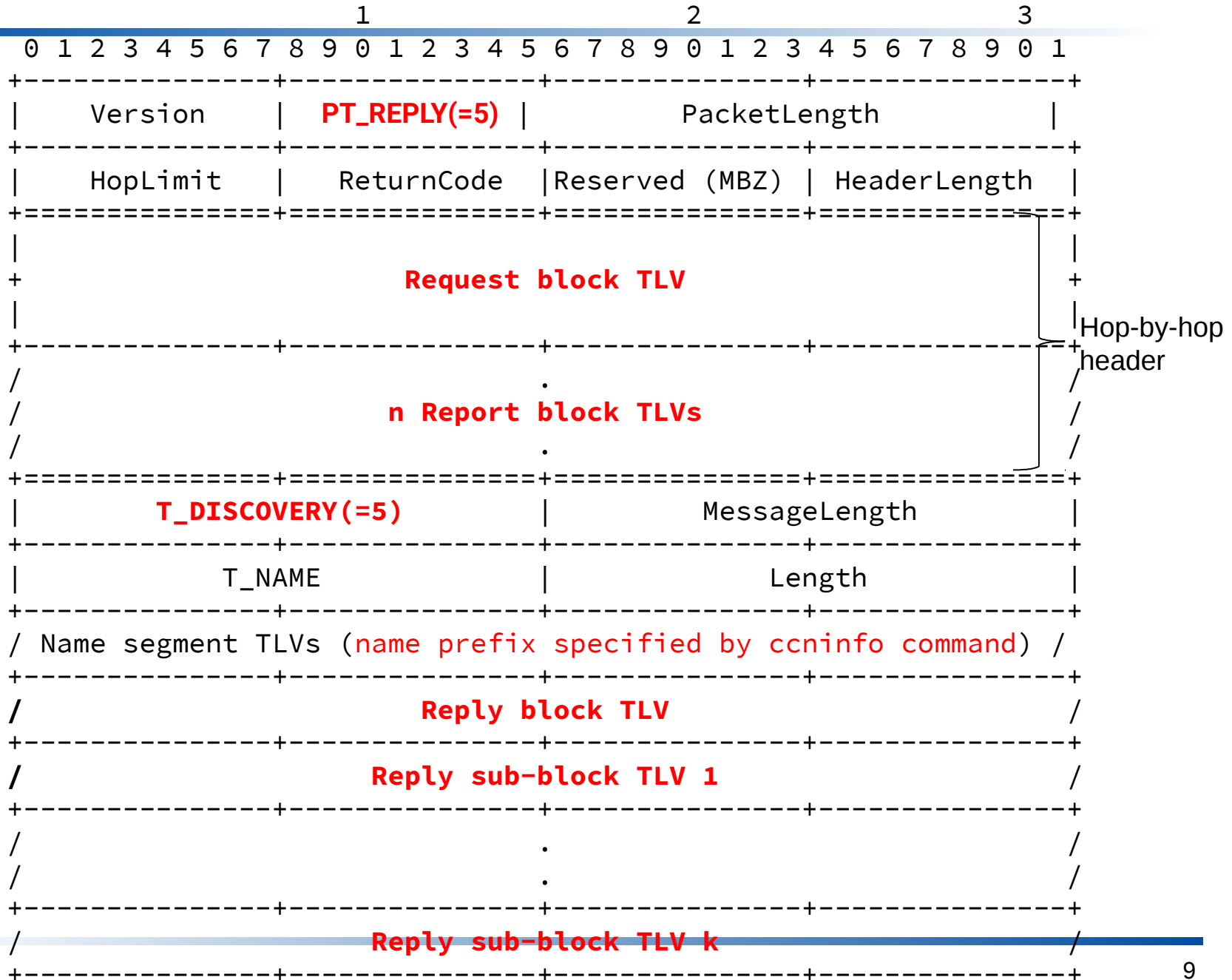


## Report block TLV

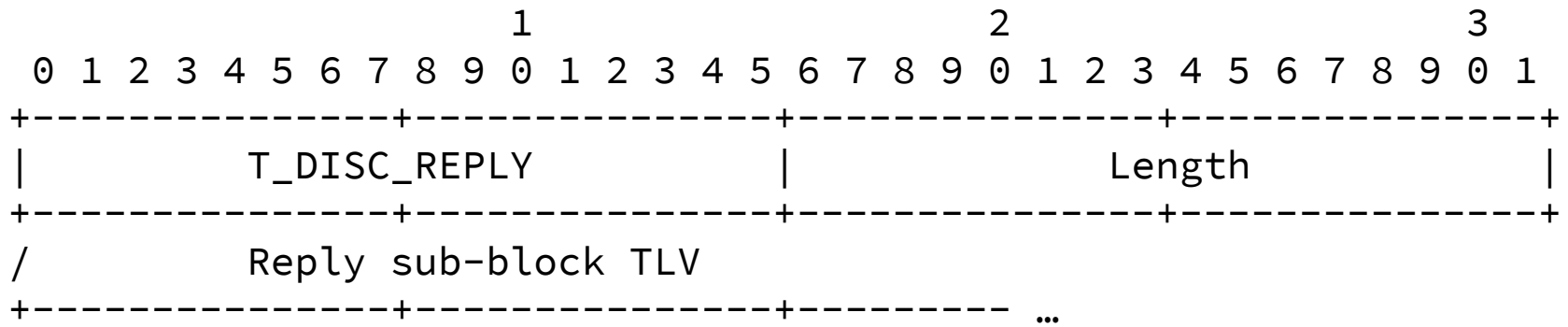




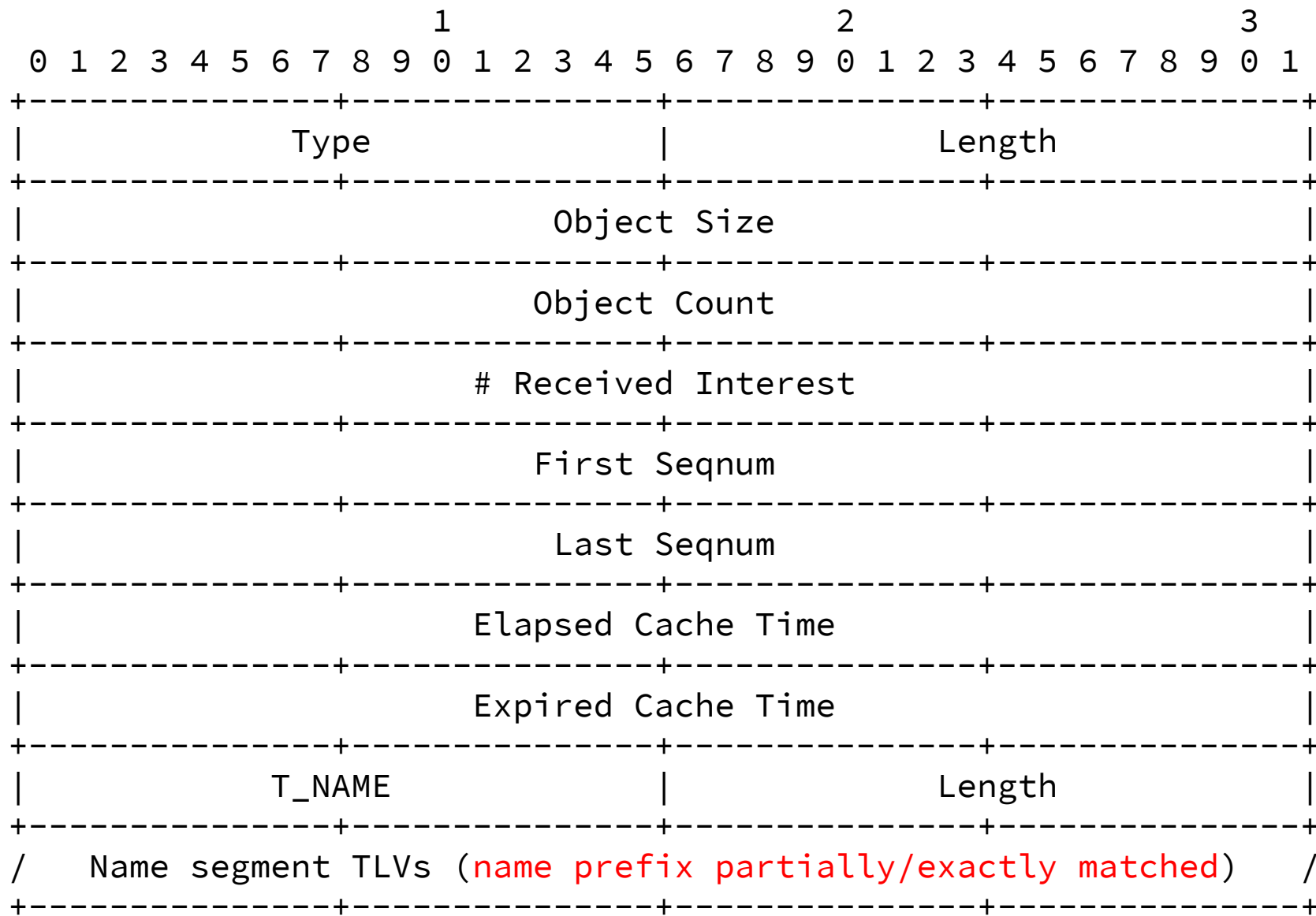
# Reply Message



# Reply Block



# Reply Sub-Block



# Unique Behavior

---

- Different operations from the regular Interest / Data are required
  - CCNinfo Requests SHOULD NOT result in PIT aggregation in routers during the Request message transmission.
  - CCNinfo Replies MUST NOT be cached in routers upon the Reply message transmission.
  - (For multipath support) Routers SHOULD NOT remove the PIT entry created by the CCNinfo Request until the timeout value expires.

# Multipath Support

- Multipath support (Section 5.2)

- The Request messages can be forwarded to multiple neighbor routers. Some router may have strategy for multipath forwarding; when it sends Interest messages to multiple neighbor routers, it may delay or prioritize to send the message to the upstream routers. The CCNInfo Request, as the default, complies with such strategy; a CCNInfo user could trace the actual forwarding path based on the strategy.

Default

- On the other hand, there may be the case that a CCNInfo user wants to discover all potential forwarding paths based on routers' FIBs. If a CCNInfo user invokes a CCNInfo Request with the parallel request flag, the forwarding strategy will be ignored and the upstream router may send Requests to multiple upstream routers simultaneously, and the CCNInfo user could trace the all potential forwarding paths. Note that this flag may be ignored according to the router's policy.

Option

# Policy-based Information Provisioning

---

- Fine-grained access control for authorized / unauthorized Requests are required, for example,
  - Router SHOULD forward the Request message to the upstream router toward the content forwarder
  - However, according to the “Node Identifier” in the Request block or “Signature” in the ValidationAlgorithm TLV, router may want to do access control and thus SHOULD be able to configure the level either of:
    - (1) All (all cache information is disclosed)
    - (2) Partial (cache information with the particular name prefix can (or cannot) be disclosed), or
    - (3) Deny (no cache information is disclosed)

# Discussions

---

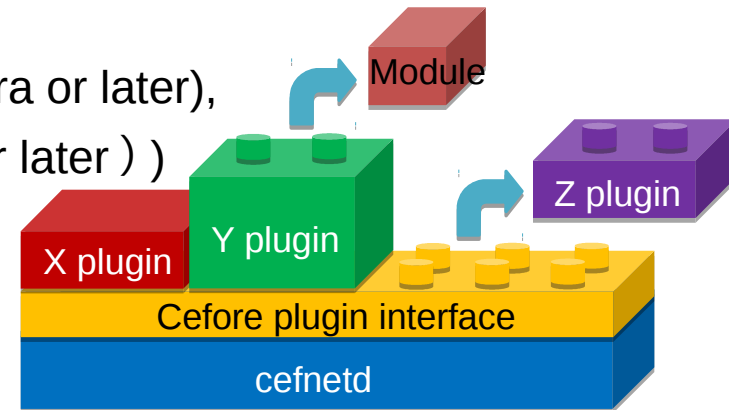
- Multipath discovery
  - PIT entry is maintained by timer so that multiple replies can be forwarded until the timer expires.
  - Another better mechanism?
- Does publisher deal with CCNinfo request?
  - Imposing such task to publishers or content servers may be difficult in some cases/situations. First-hop router attaching publisher stops Request and sends Reply message with the return code, “Reached publisher”?
- Neighbor (or adjacency) verification
  - Forwarding CCNinfo messages given from non-adjacent neighbor nodes/routers must be prohibited. Defining the secure way to verify the adjacency cannot rely on the way specified in CCNx message format or semantics.
  - We published a paper (IEEE TNSE) that includes a hop-by-hop authentication mechanism in CCN. Corresponding I-D should be proposed, or CCNinfo I-D simply refers to the paper?
- Others ...

# Cefore



- Software platform enabling CCN-based communications

- Open source written in C
  - <https://cefore.net>
- Linux (ubuntu 16.04 or later), macOS (High Sierra or later), Raspberry Pi (Raspbian Jessie), (Android ( 7 or later ) )
- Compatible with Metis (=sb-forwarder)
- Works on top of IPv4 and IPv6



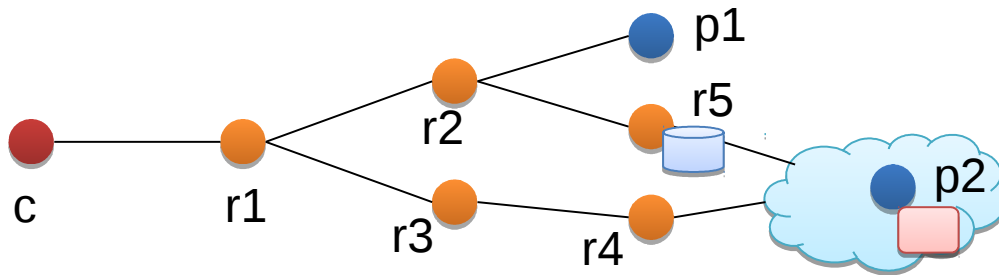
- Component

- Packet forwarding daemon ( cefnetd )
  - (1) FIB, (2) PIT, (3) Interest and Object forwarding, (4) simple CS
- Content store daemon ( csmgrd )
  - Enable one or more cefnetd(s) to share Content Store
- Plugin libraries
  - User original (or independent) libraries
    - Sample plugin: NDN packet forwarding
    - Sample plugin for csmgrd: cache replacement algorithms (LRU, LFU, FIFO, CUSH)
- Operation tools/utilities
  - cefstatus, cefroute, cefget(put)file, cefinfo (aka ccninfo), etc.



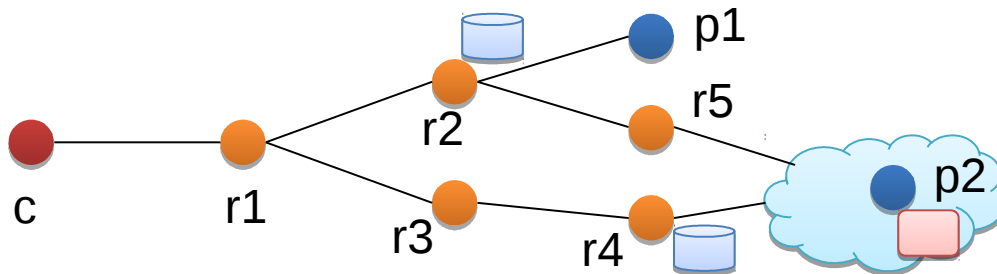
# Ex.1. ccninfo ccn:/news/science

```
consumer@ceflab: ~#  
consumer@ceflab: ~# ccninfo ccn:/news/science  
ccninfo to ccn:/news/science with HopLimit=32, SkipHopCount=0, Flag=0x0000 and Request ID=24912  
response from router5: no error, time=65.025000 ms  
  
route information:  
1 consumer          0.318 ms  
2 router1           10.165 ms  
3 router2           10.149 ms  
4 router5           11.377 ms  
  
cache information:  prefix      size      cobs      interests  start-end  expire_time  elapsed_time  
1 c  ccn:/news/science  8863 KB   6072      0  0-6071     5972 secs   26 secs  
consumer@ceflab: ~#
```



# Ex.2. ccninfo -n ccn:/video/video1

```
consumer@ceflab:~#  
consumer@ceflab:~# ccninfo -n ccn:/video/video1  
ccninfo to ccn:/video/video1 with HopLimit=32, SkipHopCount=0, Flag=0x0002 and Request ID=36407  
  
response from router2: no error, time=43.048000 ms  
  
route information:  
1 consumer          0.716 ms  
2 router1           10.151 ms  
3 router2           10.187 ms  
  
response from router4: no error, time=62.763000 ms  
  
route information:  
1 consumer          0.716 ms  
2 router1           10.151 ms  
3 router3           10.215 ms  
4 router4           10.156 ms  
  
consumer@ceflab:~#
```



# Summary

---

- CCNinfo, which is compatible with CCNx-1.0 TLV format, is a powerful network tool providing various information
- Implemented in Cefore
- Address open issues (step by step), and revise the I-D
  - Comments welcome