File-Like ICN Collections (FLIC)
draft-irtf-icnrg-flic-04
IETF 116, Yokohama

Marc Mosko
PARC

Dave Oran
Network Systems Research & Design
Outline

• What FLIC does (super quick recap)
• Minimum Requirements
• Examples with focus on NDN
  – Two Name Constructors
  – SegmentIds for unique names
  – Routing Hints
What FLIC does

- It provides a manifest of hashes that make up all the segments of a piece of application data.
- The manifest is hierarchical – that is the hash pointers can point to application data or to more manifests.
- There is a canonical traversal order. Metadata could provide other traversal hints, such as for video.
- FLIC has its own, extensible, encryption mechanism. Manifest encryption does not need to be related to content encryption.
- FLIC has several Interest construction techniques. The publisher can choose one or more of these naming techniques. More techniques could be added.
Unencrypted Manifest

Name TLV  Name  ; NDN / CCNx part
Contents TLV

Node TLV
HashGroup TLV
Ptr TLV

Hash TLV  Hash  ; FLIC part

...

Hash TLV  Hash

Signature TLV  SigInfo
NDN

• ContentType 1024 is FLIC, which serves as the “Manifest TLV” from FLIC spec.
• HashType is ImplicitSha256DigestComponent
• The Ptr Length must be an integer multiple of the PTR TLV + HashValue length.
• Because there is no NodeData, it uses the default hash-based naming.
More Practical NDN

- NDN prefers that each Data object have a unique name (not counting the implicit hash).
- FLIC uses Segmented Naming: 
  \[
  \text{common prefix} + \text{segment number} + \text{hash}
  \]
- If the manifest has one name and the application data has a second name, then one must have two HashGroups.
- One must maintain two segment number counters, one for each HashGroup.
- One must traverse the manifest in-order because there’s a global counter for each SegmentId.
Unencrypted Manifest With NodeData

Node TLV
  NodeData TLV
  NcDef TLV
    NcId TLV 0
    SegmentedSchema TLV
    Locators TLV
    Name
  NcDef TLV
    NcId TLV 1
    SegmentedSchema TLV
    Locators TLV
    Name

; Redefine the default for sub-manifest
; e.g. NDN LinkContent TLV
; Manifest name prefix
; new NcId for app data
; App Data name prefix

NcDefs are inherited between manifests. We recommend only defining them at the root.
Redefining Name Constructors

• NcDef only applies to the current manifest and its children.
• Because one cannot fetch its children before reading the current manifest, it should not cause too much burden to redefine NcDefs.
• The FLIC processor must be able to track NcDefs per manifest branch.
With NcId 0 and NcId 1

Name TLV  | Name
Contents TLV
Node TLV
   | Node Data ...
   | HashGroup TLV
   | Manifest Pointers ...
   | HashGroup TLV
      | GroupData TLV  | NcId TLV  | 1
         | App Data Pointers ...
Signature TLV  | SigInfo

; Will use NcId 0 by default
Supporting Out-of-order

• The hash pointers must include the segment number annotation.

    OR

• Give up on unique names and rely on the implicit hash to differentiate. In that case, use the PrefixSchema or other Name Constructor schema.
HashGroup with Annotations

HashGroup TLV
  AnnotatedPtr TLV
    PtrBlock TLV
      SegmentId Annotation TLV (number)
        Hash TLV Hash
        ... ; missing in FLIC-04
    SegmentId Annotation TLV (number)
      Hash TLV Hash
    ...
Alternate Proposal

One does not need to maintain a global counter. It only needs to be local to each HashGroup.
Including Routing Hints

• Routing Hints are stored under the NodeData.
• FLIC-04 calls these “Locators.”
• FLIC-04 uses the same term “Locators” for the prefix of SegmentedSchema and PrefixSchema. But in this case, these are name prefixes used in a Name Constructor.
• We should change the word in Name Constructors to not confuse the purpose.
Each FLIC packet (Node TLV) only has one set of Locators. If you need different locators for different HashGroups, you need to put only one HashGroup type per packet.
Multiple Routing Hints

• If, for example, Manifests use one set of Locators and App Data uses a different set of Locators, then each FLIC packet must be all Manifest or all AppData pointers.

• I.e., a tree where all internal Manifests only have Manifest pointers and all leaf Manifests only have App Data.
Example of single-kind manifests
Topics Not Discussed

• Encryption
• Metadata like SubtreeSize, SubtreeDigest, LeafSize, LeafDigest
• Schemas: InterestDerivedName, DataDerivedName, and PrefixName.
• ProtocolFlags
Q&A