

CCNx Technical Working Group

Meeting Minutes

11/25/15

Overview

Attendees: Jim Gibson, Dirk Kutscher, Ilya Moiseenko, Ravi Ravindran, Glenn Scott, Nacho Solis, Mark Stapp, Christopher Wood

Scribe: Christopher Wood

Agenda

1. Review and evaluate the FLIC and ISO Manifest designs and properties.
2. Discuss the proposed network-layer metadata.
3. Take comments and questions about the manifest use case document.
4. Identify volunteers for the "advanced" Manifest. (This design work will be done in parallel.).

Related Material

- FLIC notes (flic-notes.txt)
- ISO overview (iso-overview.txt)
- CCNx Message Metadata (draft-wood-icnrg-ccnxmetadata.txt)
- ICN Manifest Use Cases (draft-wood-icnrg-manifestusecases.txt)

1) FLIC notes

- What are deep trees in the context of FLIC?
 - A tree in which all entries in a (FLIC) node are data pointers except for the last one. This forms a skewed structure.
- Do (metadata) properties apply to the entire manifest or each pointer?
 - Neither. They apply to a HashGroup. Each node can have more than one HashGroup with different metadata properties. For example, pointers in one group might have different block sizes than pointers in another group.
- What's the benefit of using a tree over a list where the entries are indexed by chunk number offset? Why include hash pointers in every node?

- A skewed tree is a list. Chunking a flat (list) manifest is functionally equivalent to creating one of these skewed trees.
- Hash pointers remove the need for signature verification. Pointer hash digests are computed over the node to which the pointers refer. To verify the content object that is returned in response to a pointer, one simply computes its hash, which does not depend on the hash of the children and only the contents of the individual message, and checks for equality.
- Manifest signatures are computed over the root node to bootstrap hash-based retrieval (via transitive trust).
 - A receiver does not need to fetch the entire tree in order to verify the root signature and begin requesting child nodes (content objects or other manifests).
- Hashing the entire file will happen no matter what. How much extra hashing does the tree structure require over a list?
 - It does not require any more hashing. The number of chunks N in a list or flat manifest would equal the number of intermediate nodes N in the tree. In both cases, N additional hashes would need to be performed. The chunking approach is worse off since each chunk of the list would need to be signed as there is no hash-based reference.
- A skewed tree with a single root signature provides almost immediate application value since data can be passed upwards as soon as it passes hash-based verification.
 - Use case: a consumer wants to watch only the first 30 seconds of a 2 hour video. With the tree, they would not need to fetch the entire 2 hours worth of data before consumption.

2) ISO overview

- Main points:
 - Each pointer has a size field to enable seeking
 - Pointers can have absolute or relative names
 - Relative names, combined with the root manifest base_name, allow data to be re-named (assuming the manifest is not nameless) without re-generating entire sub-trees.
- The name in the root manifest is not the same as the base_name. The former identifies the root manifest, and the latter identifies the name to use when pulling the data which the manifest represents.

Action Items

- Provide pseudocode descriptions of the manifest tree creation and fetching strategies. [Chris]
 - Elaborate on the key design principles of FLIC and ISO. [Nacho, Chris, Christian]
 - Produce diagrams to illustrate the differences between the list (flat) and tree manifest. [Chris]
 - Show how FLIC can enable de-duping easy renaming. [Christian + Chris]
-

Questions Raised

- Is there any benefit to keeping the flat Manifest as an option?
- Will intermediate nodes in a “single object” manifest ever contain names? Should they?

Next Meeting

Date & Time: 12/9/15 at 11am PST

Tentative agenda:

- Continue the FLIC and ISO design discussions.
 - Discuss the ICN Manifest Use Case document and its impact on the “advanced manifest.”
 - Identify a volunteers or a task force to work on the “advanced” manifest.
-