CCNx Technical Working Group

Meeting Minutes

12/9/15

Overview

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

Scribe: Christopher Wood

Agenda

1. Continue discussion and evaluation the FLIC and ISO manifest designs, with a focus on de-duping and sizes per pointer.
2. Propose changes to the CCNx Semantics and Messages documents.
3. Briefly revisit interoperability barriers (link protocol, forwarder discovery, etc.) and identify the gaps.
4. Briefly discuss advanced manifest use cases.
5. Identify volunteers for the "advanced" manifest. (This design work will be done in parallel.)

Related Material

- FLIC vs ISO email (naming-notes.txt)
- FLIC specification (https://github.com/tschudin/icn-flic-rfc/blob/master/draft-tschudin-icnrg-flic-00.txt)
- ISO overview (iso-overview.txt)
- CCNx Semantics (https://www.ietf.org/id/draft-irtf-icnrg-ccnxsemantics-00.txt)
- CCNx Messages in TLV Format (https://www.ietf.org/id/draft-irtf-icnrg-ccnxmessages-00.txt)

1) FLIC and ISO Discussion

- (Pro-ISO claim): organization by groups does not necessarily help.
  - Does not save much more space.
  - Each one has efficient and inefficient scenarios.

    - Relative names can be sacrificed.
- Gave implementation status update.
    - FLIC implemented in ccn-lite and CCNx.
    - Repo implemented in ccn-lite but not in CCNx.
- How does the FLIC design interoperate with the encrypted manifests, file system-like seeking, and cleartext names?
    - Balanced manifest trees enable file seeking to arbitrary blocks in the data (across the network).
- Manifest does not solve the problem of "what byte to seek to in order to get the right video?" It's just a low level index for a large object.
- FLIC receivers need to do bounds checking for a "seek" due to hierarchical manifests with possibly different child block sizes.
- Do these manifests subsume the chunking protocol?
    - Unsure, but it can, and also maybe fragmentation if we have that knowledge
- How are names used to index into the repo?
    - Names are locators for the correct chunk and the hash is the unique identifier.
- Should multiple locators be maintained in the root? And what are the security and trust implications of that addition?
- Third possible de-duplication location: cache based on hash instead of name and hash (to satisfy multiple interests with different locators and the same hash restriction).
- Encryption and manifest are not necessarily independent: what if a person sees a cleartext hashes and then uses them to subvert the root manifest to get access to chunks.
    - Should match by hash be the right thing to do in a forwarder?
    - Do names subsume the interest name locator in the forwarder?
- Repos have different matching semantics from forwarder cache.
- Nameless objects are unsettling -- do messages really not need to contain the name?
    - Is it an important capability to retrieve by hash only without anything in the name?

Action Items

- Formalize nameless object matching, forwarding, and repo rules. [Chris]
- Explore the relationship between manifest encryption and nameless object fetching. [All]

Questions Raised

- Is chunking still necessary with manifests?
- Is matching based on hashing alone (i.e., not taking the name in account) a security problem?

Next Meeting

Date & Time: 12/23/15 at 11am PST

Tentative agenda:

- Manifests, encryption, and nameless objects
- Nameless object forwarding
- FLIC updates