

# CCNx Technical Working Group

## Meeting Minutes

2/3/16

## Overview

---

**Attendees:** Dirk Kutscher, Börje Ohlman, Ravi Ravindran, Greg Rutz, Glenn Scott, Nacho Solis, Mark Stapp, Christopher Wood, and others

**Scribe:** Christopher Wood

## Agenda

---

1. CCNx code release status update (<http://blogs.parc.com/ccnx/ccnx-downloads/>)
2. Interoperability: CCNx over UDP, forwarder discovery, link negotiation (ICN-LLC).
3. Manifests and transport protocols (time pending).

### 1) Code Release Status Update

---

- Released code on Github and are currently going through the new development workflow.
- Overview:
  - Code is distributed as separate modules (e.g., LongBow, Libparc, Libccnx).
  - Transitioned from automake to CMake.
- Best mailing list to use for questions and concerns is [ccnx@ccnx.org](mailto:ccnx@ccnx.org).
- Filing issues in Github is a good (the best?) way to report and track problems.
- A contribution README would be helpful on Github to enable people to collaborators to participate.
- Q: What is the roadmap?
  - It's open to the community right now as things progress.

### 2) Interoperability

---

- Need to revisit interoperability -- we have work on "CCNx over UDP" and forwarder discovery.
- Q: Is naming part of the interop task?
  - It's mainly orthogonal will need to be dealt with soon.
  - It's beyond the base interop case we're discussing here.
- We need to focus on getting two forwarders to talk to one another (the bottom most layer, or the minimal set of functionality).
- Q: Adjacency establishment is sometimes part of the routing protocol. Is that the case here?

- Currently, no.
- Reiterating past sentiment: we should be built on top of existing technologies (e.g., UDP and DTLS).
- Interop techniques are currently separate (discovery is like plugging a cable into a box)
  - The “CCNx over UDP” draft describes what is necessary to establish a security association. This requires the forwarder discovery technique to identify the right forward to whom to connect.
- Q: How many layers are in this interop stack for the baseline?
  - If it’s just UDP, we can rely on UDP(+DTLS) for it to work.
- Q: Do we not need a way to negotiate or state assumed parameters for the transport mechanism (e.g., UDP MTU)?
  - Maybe not. We should rely on the common case and omit unrealistic cases (e.g., a UDP link with a 64K MTU).
- We should be explicit about the rationale about restrictions on the link protocol (e.g., what’s the reason we chose a 4k MTU for the protocol, and why packets don’t traverse secure links)?
- Q: What if we want the link to be delivered in order?
  - UDP doesn’t give you that (nor does Ethernet, end-to-end).

### 3) Manifests and Transport Protocols

---

- The transport protocol would use the manifest (which is structured data) for execution.
  - We need to iron out the last remaining details of the manifest (FLIC) first.
  - We (the community) would benefit from an exploration of manifest usage in existing applications (e.g., how would Flickr use Manifests?).
    - This would help us learn where and how it’s applicable for applications.
  - Mark Stapp is working on an exploration on mapping real-world applications to a CCN world.
  - Comment: Experimentation would highlight many of the real-world use cases and problems.
-

## Action Items

---

- Outline interoperability steps. [Chris]
- Application architecture and privacy requirement exploration. [Chris and Mark]
- Experiments. [All]

## Next Meeting

---

**Date & Time:** 2/17/16 at 11am PST

**Tentative agenda:**

- Review the latest interoperability documents.
  - Review Manifest use cases for applications and outline experiments to be done.
  - Break down a single “real world” application.
-