Antitrust Policy Notice

Linux Foundation meetings involve participation by industry competitors, and it is the intention of the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.

Examples of types of actions that are prohibited at Linux Foundation meetings and in connection with Linux Foundation activities are described in the Linux Foundation Antitrust Policy available at http://www.linuxfoundation.org/antitrust-policy. If you have questions about these matters, please contact your company counsel, or if you are a member of the Linux Foundation, feel free to contact Andrew Updegrove of the firm of Gesmer Updegrove LLP, which provides legal counsel to the Linux Foundation.

Hyperledger is committed to creating a safe and welcoming community for all. For more information please visit our Code of Conduct: Hyperledger Code of Conduct.

Attendees

- David Fuelling
- Ian Simpson
- Neil Hartner
- Noah Kramer

Agenda

1. Introductions
2. Discuss open Quilt issues & PRs
3. Q&A, misc issues

Goals

- Stakeholder sync-up.

Discussion items

<table>
<thead>
<tr>
<th>Time</th>
<th>Item</th>
<th>Who</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 min</td>
<td>Intros</td>
<td>David Fuelling</td>
<td></td>
</tr>
</tbody>
</table>
Discuss project issues

- Discuss StreamRFC PRs (https://github.com/interledger/rfcs/pull/573).
  - Quilt PR: https://github.com/hyperledger/quilt/pull/459
- Neil: anecdote about the “refer” keyword in the HTTP spec that was specified as incorrect because this how everyone was doing it.
  - Maybe a ConnectionAssetDetailsRequest that can trigger the other side to send its CAD frame.
- Codec Failure Validation: https://github.com/hyperledger/quilt/pull/456
- STREAM sender calculations:
  - Max amount
  - Min amount
  - Dust
  - Slippage
  - Connection failure statistics.
- Durable STREAM Sender
  - SimpleStreamSender is not simple.
  - **Goal**: Update a datastore to say “this is how much you sent” and Bisected Packet detection.
  - Bisected Packet Detection: In large-packet-amount world, you want to say, “I’m about to send a packet for 1000 units” so you can check it later in the event of a server crash.
  - Introduce a filter-chain
    - 1.) Help to figure out what the payment amount should be.
    - 2.) Persist the packet.
    - 3.) Log the packet, etc.
    - 4) On response, do other things.
  - Parallelization of Packets introduces complexity
    - Current impl starts small and ramps up.
    - New Impl will start large and ramp down.
  - Try single-threaded simpler stream sender.

15min

Action items

- Upload call audio to wiki and link here. David Fuelling

Recordings

audio_only.m4a