CCN/NDN Protocol Wire Format and Functionality Considerations

G.Q.Wang¹, Christian Tschudin², Ravi Ravindran¹
(ICN-RG, Hawaii, IETF-91)
[1] Huawei, Santa Clara, USA
[2] University of Basel, Switzerland

Nov 2014, v1.5
Agenda

• Motivation
• Multiple TLV Schemas
• Elastic TLVs
• Forwarding Target
• Header Compression
• Selectors Usage
• Non-Shareable Content
• Caching as a service
• Context Handling
• Summary
Motivation

• TLV discussions are primarily considering performance requirements.
• Future Internet Architecture has to accommodate other requirements too.
  – Flexibility
  – Scalability
  – Expressiveness
    which needs support at the wire format level
• Following are some of these requirements for future considerations.
Flexible TLV Schema(s)

• “One TLV to rule them all” is bad. Need support for a multiplicity of TLV schemas:
  – one (or few) TLV format for the fixed header
  – potentially many TLV flavors in the option fields and payload
    (policies might restrict what a net accepts, but the functionality is very useful)

• Examples:
  – To support Backward Compatibility and Service Expressiveness
  – forward a CCNx2.0 payload through a CCNx1.0 net
  – forward a NFN thunk [1] representation through CCNx1.0
  – Service composition [2]

• Relies on a generalized “name-to-forward-on” schema, see the “forwarding target pointer” slide later on

Elastic TLV for CCN

Variable “Length” definition to accommodate heterogeneous application/device/interface-capability contexts e.g. Optical, IoT

• One possibility to support large PDUs

<table>
<thead>
<tr>
<th>T= 2B</th>
<th>Flag Bits (2b)</th>
<th>L= 14 bits</th>
</tr>
</thead>
</table>

(00) B/Unit-Size
(01) KB/Unit-Size
(10) MB/Unit-Size
(11) GB/Unit-Size

• The proposal keeps it simple, in terms of limiting over head to 2/2 Type and Length, while using two bits to determine granularity of the payload.

• The selection of the per-unit resolution can be chosen by the application, based on the feedback from ICN forwarding layer, based on strategic path level feedback.
Forwarding Target Pointer (a.k.a Locator)

- Allow **Interest forwarding** to operate on something other than the Interest name proper (which nevertheless stays in the packet)
- ICN Name, or Flat Label, or ...
  - `/huawei/g.q/phone` ➔ `/att/sc/ap-x` [1]
  - alternate name or flat label for mobility mechanisms like Kite [2]
- Supports mobility, late-binding, or other application-centric requirements.
- **Proposal** (examplified for CCNx1.0):
  store the name bits, as well as the pointer, as optional hdr TLVs

<table>
<thead>
<tr>
<th>Header</th>
<th>Forwarding-Target-Pointer Bytes (Optional TLV)</th>
<th>Forwarding Label (Optional TLV)</th>
<th>Interest Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>[FT-flag]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **An FT-flag** indicates the presence of a Forwarding Target Pointer. The first optional hdr field MUST be the FTP (quick access at fixed pos.)
  \[
  T=\{\text{Forwarding-Target-Pointer}\} \quad L=\text{sizeof(offset)} \quad V=\{\text{offset-of-"FT-Bytes"}\}
  \]

- The name or label bits can be anywhere in the optional header field area:
  \[
  T=\{\text{Name-or-Label-Type}\} \quad L=\text{sizeof(name-or-label)} \quad V=\{\text{name-or-label bytes}\}
  \]

- **Hdr-Len field is still used to access the payload (and its Name-TLV)**

Header Compression

• Hooks for header compression, especially for names. But *encoding context switching* could also be used for type dictionaries as in ccnb.

• **Others do it too:**
  Remember MNP5 from old modem times[1], TCP header compression, UDP ROHC [2], and 6LoWPAN?

• Examples:
  - Ask downstream node to accept “name abbreviations”. The name mappings would be stored in a “context”, hence the need for a “contextID” field in the fixed header.
  - IoT setting: use a 1+1 TLV schema internally, the gateway will expand it to 2+2 for the rest of the world.

[2] RFC 1144, RFC 2058, RFC 4019 (Robust Header Compression)
Caching as a Service

• CCN/NDN domains may not have any caching at all.
• Or domains could enable caching/storage only at the edges.
• Recent PARC document [1] on distributing PIT/CS and FIB functionality.
• Introduce packet processing complexity where it is more useful.

Shareable versus Non-Shareable

• Non-Shareable content (e.g. conversational, transactional) can be on fast path without PIT/CS processing.
  – As communication is bi-directional, optional source-ID can be included.

• As Optional Header TLV

| Header [FF-Flag] | Interest Name | Source-ID |

Using Selectors

• Selectors as a Optional feature.
  – Implication on the PIT design

• Selectors can be avoided in the network infrastructure with authoritative sources exist.

• Selectors are useful where authoritative source doesn’t exist, and learning from cache or source is the only option.
  – Discovery Services, Inventory in Home, Campus etc.
  – Ad hoc V2V, IoT scenarios

• Should be a Protocol Feature that can be optionally enabled
Context Handling

- Provision to include context metadata that can be processed in the Network Layer.
  - Contexts includes Identity/Location/Device etc.
  - Attachment to a Service Instance
  - Discovering Content/Services
  - Policy based Routing/Forwarding
  - **Optional Interest TLVs**

| Header | Interest Name | {Context Metadata} |
Summary

• CCN/NDN Protocol design not just on performance, but also on flexibility, scalability, and expressiveness.

• Several considerations laid out to be accounted for current design and future enhancements.

• Eventual consensus between CCN and NDN, do not desire two versions of the same protocol.