Support for Notifications in CCN (“draft-ravi-ccn-notification-00.txt”) IETF/ICN-RG -93, Prague

Ravi Ravindran (ravi.ravindran@huawei.com)
Asit Chakraborti(asit.chakraborti@huawei.com)
Marc Mosko(marc.mosko@parc.com)
Ignacio Solis(ignacio.solis@parc.com)

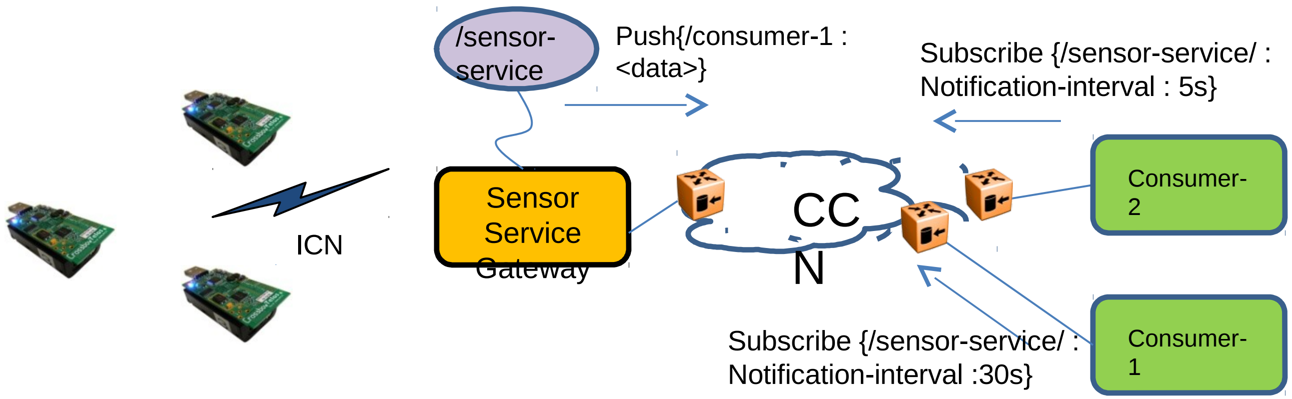
Need for Notification

- Notification provides asynchronous way to update listeners about a particular event related to:
  - Periodic updates from a monitoring system
  - Event triggers due to state change of a control system
  - Command operations in actuating situations
  - Discovery when nodes come online
  - Control Plane Interaction due to state or context change
- In such situations PUSH is more energy efficient than PULL or Probing.
- Notifications can be Unicast or Multicast
- Its use pervades many application such as in IoT and Social Networking
- Notification traffic type varies from being mission critical to best effort.
  - Hence reliability can be at the level of application, transport, or the network layer
- The latency requirements can be very stringent, ~5ms [1] for certain systems.
  - E.g. Traffic Systems, Remote Surgeries etc.

[1] Osseiran. A et al“Scenarios for 5G mobile and wireless communications: the vision of the METIS project”, IEEE Communication Magazine, 2014

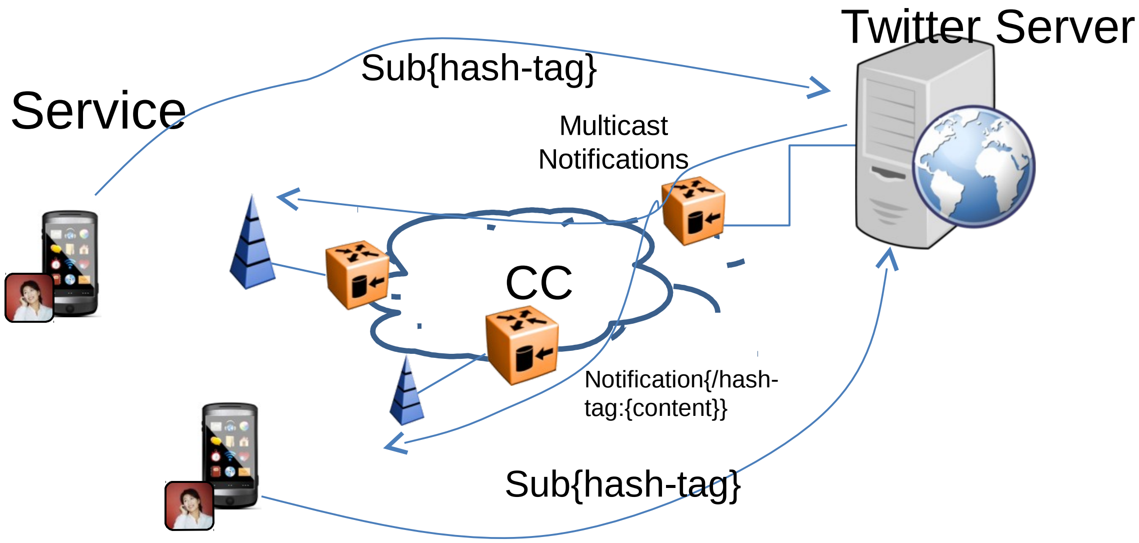
Notification Use Cases

- Subscribing to some sensor information


/sensor-service
Push{/consumer-1 : <data>}
Subscribe {/sensor-service/ : Notification-interval : 5s}
ICN
Sensor Service Gateway
CCN
Consumer-2
Consumer-1
Subscribe {/sensor-service/ : Notification-interval :30s}

IoT Scenario

- PUB/SUB Service



Social Networking

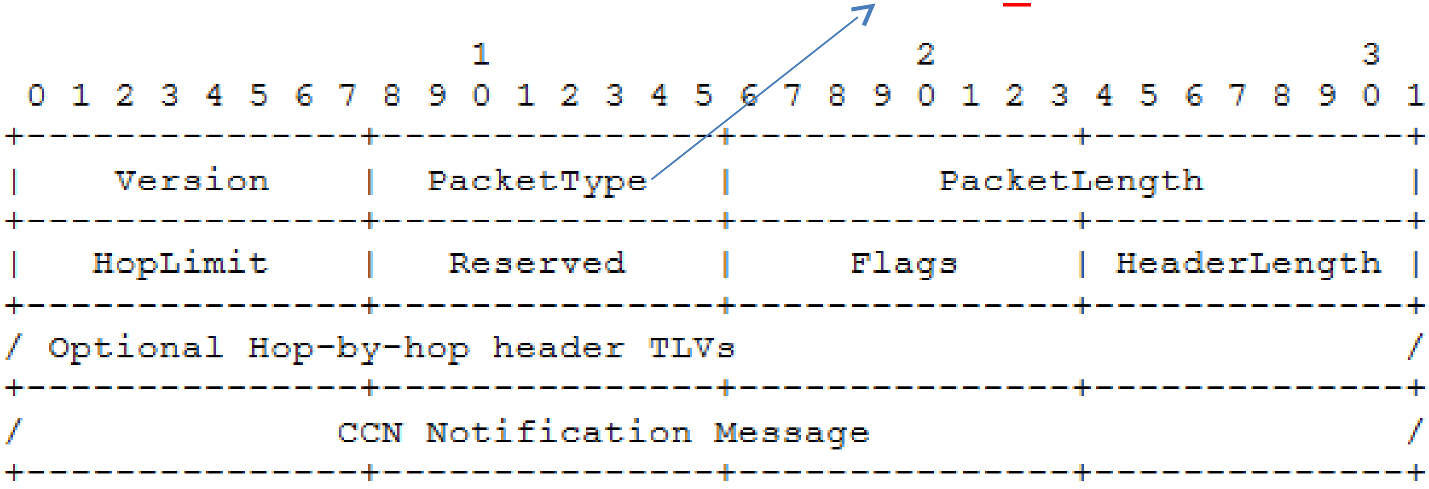
Current Support for Notifications in CCNx1.0

- Current support is by emulating PUSH though the Interest Pull API
- In CCNx1.0, the interest life time can be set to 0, to avoid state at the intermediate forwarders.
- Several Issues:
  - Depending on implementation, the Interests can be subjected to PIT/CS processing which incurs computing and latency cost.
  - Using Interest life time to indicate this important type of traffic class is not desirable.
    - How to differentiate between legit PULL with zero lifetime from PUSH ?
  - Differentiating among different types of PUSH traffic, e.g. mission critical PUSH versus Best-effort.
  - Forwarder should differentiate between routing policies applied to PUSH versus PULL.
    - Differentiating PUSH/PULL traffic aggregating under the same prefix in the forwarder.
    - E.g. PUSH forwarding requires simple multicast, without any strategy layer intervention.
  - In an PUSH emulation scenario : How to carry Content Objects in an Interest to take advantage of its features like cache lifetime, secure name-data binding?

Notification Proposal considering CCNx1.0

- Notification is identified with a new transport primitive in the fixed header.
- Here PacketType is set to TYPE_NOTIFICATION.
- Allows forwarder to apply special packet processing and routing/forwarding logic.
- When forwarder encounters this type of traffic, only FIB state in the forwarder should be used.
- New hop-by-hop fields relevant to Notifications.

TYPE_NOTIFICATION



```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---------------+---------------+---------------+---------------+
|    Version    |  PacketType   |         PacketLength          |
+---------------+---------------+---------------+---------------+
|   HopLimit    |   Reserved    |     Flags     | HeaderLength  |
+---------------+---------------+---------------+---------------+
/ Optional Hop-by-hop header TLVs                               /
+---------------+---------------+---------------+---------------+
/                 CCN Notification Message                      /
+---------------+---------------+---------------+---------------+
```

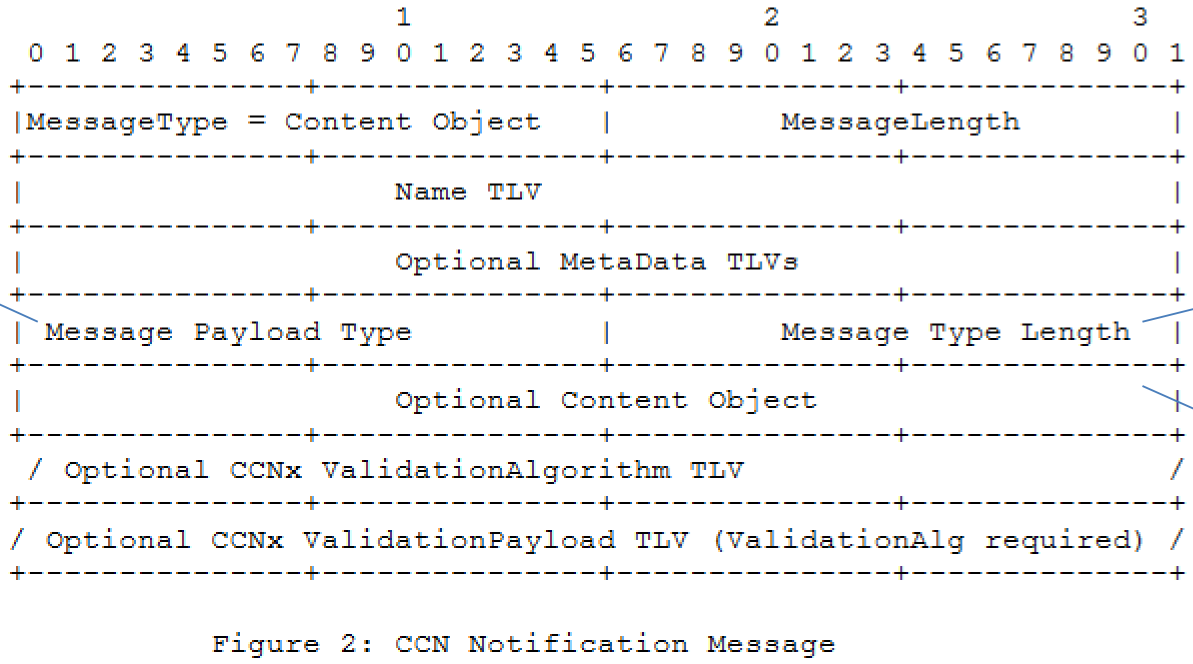
Figure 1: Notification fixed header

Notification Message Considering CCNx1.0

- CCN Notification message is a Content Object, which can optionally encapsulate another Content Object.
- Top level CO Name TLV used for forwarding.
- The **Message Payload Type** optionally includes a new T_ENCAP type payload which optionally encapsulates another CO.
  - This separates Routing Namespace from Content Producer Namespace
  - With only top CO, the Consumer/Producer agrees on the same namespace for routing and producing Content.
- **The draft requires these Content Objects to be not cached in the network.**

Four type
- Data
- Link
- Manifest
- Key

-> T_ENCAP (New)



```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---------------+---------------+---------------+---------------+
|MessageType = Content Object   |         MessageLength         |
+---------------+---------------+---------------+---------------+
|                          Name TLV                             |
+---------------+---------------+---------------+---------------+
|                   Optional MetaData TLVs                      |
+---------------+---------------+---------------+---------------+
| Message Payload Type          |           Message Type Length |
+---------------+---------------+---------------+---------------+
|                    Optional Content Object                    |
+---------------+---------------+---------------+---------------+
/ Optional CCNx ValidationAlgorithm TLV                         /
+---------------+---------------+---------------+---------------+
/ Optional CCNx ValidationPayload TLV (ValidationAlg required) /
+---------------+---------------+---------------+---------------+
```

Figure 2: CCN Notification Message

T_PAYLOADTYPE_LEN

T_PAYLOAD TLV

Security Implications

- **Flow Balance**
  - Current model manages flow balance in the network with 1:1 relationship between an expressed Interest and returned CO.
  - Unsolicited CO transmission over a CCN infrastructure violates this principal.
  - Cannot throttle traffic at the edges which CCN enables using PIT.
- **Cache Poisoning**
  - No caching recommendation of this draft
  - But it is open research to understand policy based caching implications of these notification objects to increase data availability
- **Other issues:**
  - Require mechanisms to handle End-to-end Reliability, Flow and Congestion Control for Notifications.
    - “draft-ietf-core-observe-16” has several considerations on this regard in the context of CoAP protocol.
  - Size of the allowed Content Object
    - Possibly impose restriction on the size of Notification, forwarder may drop beyond this size.

Conclusions

- The draft proposes a new Notification primitive for CCN.
- This allows forwarder to apply new processing logic to this new traffic type.
  - Avoiding PIT/CS processing
  - Notification specific Routing/Forwarding Policies
- The notification CO shouldn’t be cached.
  - Should be investigated as a possible feature to increase data availability.
- CCN Notifications have implications on flow control, Caching, and end-to-end reliability which require more research.