Support for Notifications in CCN (“draft-ravi-icnrg-ccn-notification-00.txt”)

IETF/ICN-RG -96, Berlin

Ravi Ravindran (ravi.ravindran@huawei.com)
Asit Chakraborti(asit.chakraborti@huawei.com)
Syed Obaid Amin (obaid.amin@huawei.com)
Jiachen Chen (jiachen@winlab.rutgers.com)
Marc Mosko(marc.mosko@parc.com)
Ignacio Solis(ignacio.solis@parc.com)

Draft ToC

- Introduction
- Notification Requirements
- Current Approaches
- Proposed Notification Primitive
- Notification Message Encoding
- Notification Processing
- Security Considerations
- Annex
  - Flow and Congestion Control
  - Routing Notifications
  - Notification Reliability
  - Use Case Scenario
    - Pub/Sub System

Motivation for PUSH in CCN

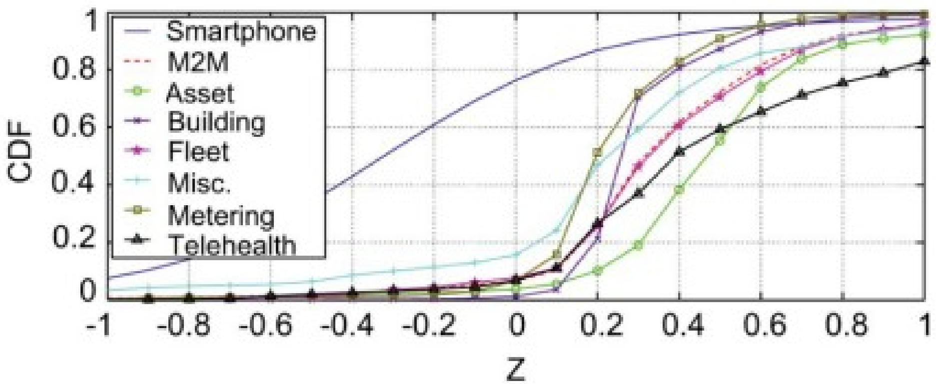


Fig. 1: Log Ratio of Upstream to Downstream traffic for M2M and Smart Phone

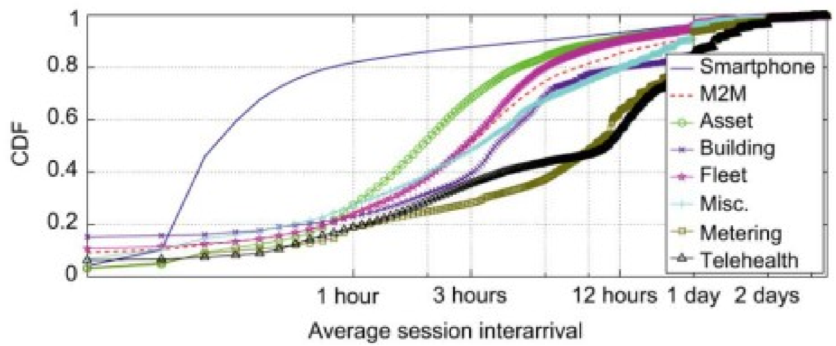


Fig. 2: Distribution between transmission range from hours to days.

- From Fig 1., significant (>80%) number of M2M devices have traffic that is upstream heavy.

- From Fig. 2, the distribution between the transmission vary from mins to days. Some of these updates are mission critical, so cannot be afforded to be lost.

- 5G puts IoT as a game changer for the industry, hence proposals such as Network Slicing [2]

[1] Shafiq et al, “Large scale measure and characterization of cellular machine-to-machine traffic”
IEEE, Transactions on Networking, 2013
[2] **ITU, FG, IMT 2020 – “Network Standardization Requirement for 5G”**
http://www.itu.int/en/ITU-T/focusgroups/imt-2020/Documents/T13-SG13-151130-TD-PLEN-0208!!MSW-E.docx

CCN PUSH Requirements

- **Supporting PUSH Intent**
  - This should match application’s intent to PUSH content similar to the PULL primitive.
- **Support Multicast**
  - Support network service where an application PUSH can be multicasted to **all** intended receivers (just like Interest Multicast)
- **Security**
  - Should be able to deliver secure (authenticated and encrypted) content objects
- **Routing and Forwarding Support**
  - Push prefixes (Multicast or Unicast) should be treated differently from prefixes for regular Interests from routing and forwarding perspective.
- **Minimizing Processing**
  - PUSH flows shouldn’t be subjected to PIT/CS processing, considering latency and application intention.

Current Approaches

- **Polling**: Consumer should periodically check for new content.
  - Pros
    - Leverage current Interest/Data primitive
  - Cons
    - Here Consumers work with partial names (the filters have to be set appropriately), could get stale cached content.
    - Potential loss of critical events
    - Additional processing and states if updates have really long intervals
    - Choosing the frequency of polling is tricky
      - high frequency leads to overhead, and low frequency in inefficiency.
    - Doesn’t meet Multicast requirement.
    - Not suitable for Constrained Producers
- **Long Lived Interests**: This is a variation of Polling, with Interests set to long lifetime
  - Pros
    - Multicasting can be supported with this, albeit with high PIT cost
  - Cons:
    - Has to work with Partial names.
    - The concerns here are regarding choosing the right lifetime and efficiency of the solution.
    - This approach is generally not encouraged

Current Approaches

- **Interest Overloading:** Here the Interests are suffixed with more parameters to convey notification messages.
  - Pros
    - Can Support Multicast
  - Cons
    - Difficult to convey Secure Content Objects - Interest Payload is a better approach.
    - Interests put through PIT/CS processing, also treated as content fetch
    - Mission critical Notifications can be blocked by other Interests (potential attack) (depending on the rules for Interest matching)
- **Interest Trigger:** Uses an Interest to trigger a PULL from the Applications
  - Pros
    - Support Multicasting/Secure Content Objects Push.
  - Cons
    - Higher latency : additional RTT compared to previous approaches
    - Mission critical application suffers with this approach

Notification Proposal

- Notifications is a new CCN primitive to PUSH Content Objects to listening consumers. In term of stated requirements.
  - It is to support Applications PUSH intent
  - Supports multicast using FIB
  - All the Security features of Content Object is preserved
  - Control/Forwarding plane can be incrementally upgraded to distinguish these prefixes.
  - Notification flows can be identified, and made to by pass PIT/CS processing

Notification Proposal considering CCNx1.0

- Notification is identified with a new transport primitive in the fixed header.
- Here PacketType is set to NOTIFICATION.
- Allows forwarder to apply different packet processing and routing/forwarding logic.
- When forwarder encounters such flows, only FIB state in the forwarder should be used ; doesn't involve PIT/CS processing.
- New hop-by-hop fields relevant to Notifications can be included (e.g. requiring ACK)

```
                     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=  |                               |
|               | Notification  |          PacketLength         |
+---------------+---------------+---------------+---------------+
|   HopLimit    |   Reserved    |     Flags     | HeaderLength  |
+---------------+---------------+---------------+---------------+
/               Optional Hop-by-hop header TLVs                 /
+---------------+---------------+---------------+---------------+
/            Content Object as Notification Message             /
+---------------+---------------+---------------+---------------+

          Figure 1: CCN Notification fixed header
```

Notification Message Considering CCNx1.0

- CCN Notification protocol message is a Content Object, which can optionally encapsulate another Content Object.
- Top level CO Name (outer CO) TLV used for forwarding.
- The **Message Payload Type** optionally includes a new T_ENCAP type payload which optionally encapsulates another CO.
  - This is required if the producer is operating on a different name space
- **The draft requires these Content Objects to be not cached in the network.**

```
                     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   |
+---------------+---------------+---------------+---------------+
|                    Payload or Optional Content Object         |
+---------------+---------------+---------------+---------------+
/          Optional CCNx ValidationAlgorithm TLV                /
+---------------+---------------+---------------+---------------+
/ Optional CCNx ValidationPayload TLV (ValidationAlg required) /
+---------------+---------------+---------------+---------------+
```

Figure 2: CCN Notification Message

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.
- **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 multicast efficiency, reliability, 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.

Annex: Flow and Congestion Control

8.1. **Flow and Congestion Control**

- 8.1.1. Issues with Basic Notifications
- 8.1.2. Flow and Congestion Control Mechanisms
  - 8.1.2.1. End-to-End Approaches
  - 8.1.2.2. Hybrid Approaches
- 8.1.3. Receiver Reliability

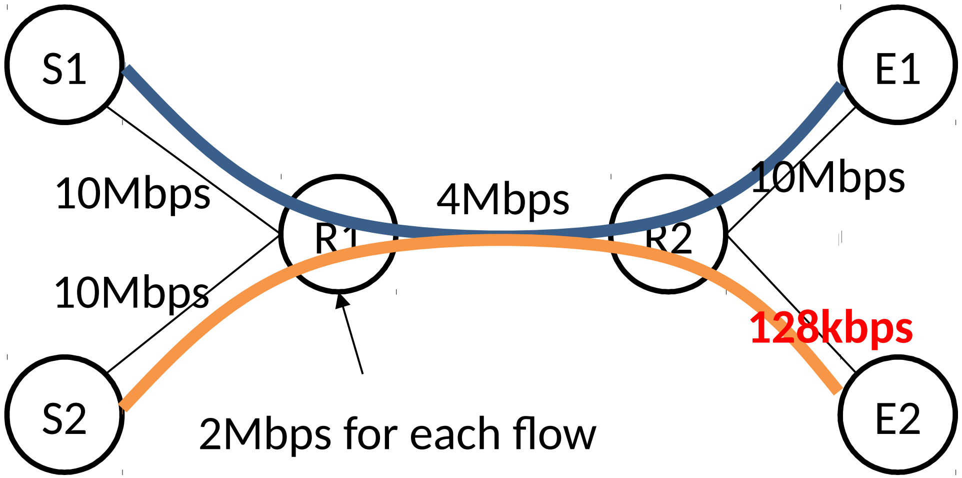
Issue with Basic Notification

- Without any flow balance mechanisms, senders could overwhelm the receivers and the network capacity.
- Two basic approaches
  - End to end Approaches
    - Receiver/Sender Oriented
  - Hybrid Based
    - Could leverage caches in the network

Pure Network-based Solution

- Flow-based fair queueing
  - Scalability issue: amount stored in the routers
  - Efficiency issue: under-utilize available bandwidth

Ideal throughput: 4Mbps
Now: <2.2Mbps



- There is a need for feedbacks from the receiver end

Endhost-Driven Solutions

- **Sender-Driven Multicast**
  - Since the Notification in ICN is very much like the multicast in IP, similar ideas can be adapted here.
    - E.g., pgmcc[1], TFMCC[2]
  - Avoid ACK (feedback) implosion
    - Feedback between the sender and only one receiver (pgmcc)
    - Measure RTT to calculate TCP Friendly Rate (TFMCC)
  - Ensure network fairness
    - Single rated, therefore the whole flow (no matter which branch) should be friendly to other flows
    - Naturally, picks the slowest receiver
  - **Issue:**
    - Throughput can be affected by a slow receiver

[1] Rizzo, L., "pgmcc: A TCP-Friendly Single-Rate Multicast Congestion Control Scheme.", SIGCOMM CCR vol. 30.4, pp. 17-28, 2000,
[2] Widmer, J. and M. Handley, "TCP-Friendly Multicast Congestion Control (TFMCC): Protocol Specification.", IETFRFC 4654, 2006.

Endhost-Driven Solutions

- **Receiver Driven**
  - E.g., RLM [1]
  - Sender generates multiple groups (name prefixes) with different rates
  - Receiver joins a certain group according to his available bandwidth/resource
  - **Issues:**
    - Complicated logic on the sender
    - Coarse-grained congestion control result in under-utilize the network

[1] McCanne, S., Jacobson, V., and M. Vetterli, "Receiver-driven Layered Multicast.", SIGCOMM CCR pp. 117-130, 1996.

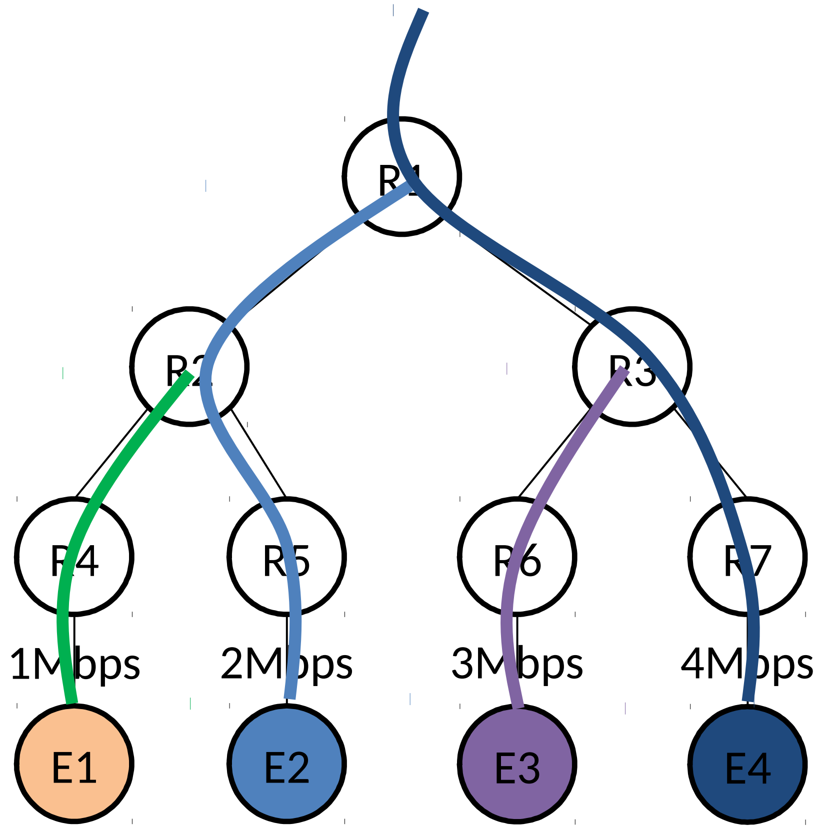
Hybrid Solutions

- **Category 3: Network Assisted**
  - Raises an important issue with AIMD and Heterogeneous receivers, and network caching.
  - Endhost sends out subscriptions for only 1 packet
    - Can also see it as an Interest for “*any next* coming packet” under a prefix, avoid cache lookup
  - Endhost maintains a congestion window
    - Not unlike the congestion control query/response
  - The routers maintains a count for each flow, each face
    - +1 on receiving the subscription
    - -1 on forwarding a Notification
    - Update max count upstream

[1] J. Chen, M. Arumaithurai, X. Fu, K.K. Ramakrishnan “SAID: A Control Protocol for Scalable and Adaptive Information Dissemination in ICN" *Proceedings of the 3rd (Also on ArXiv)*

Hybrid Approach

- Category 3: Network Assisted
  - Proof of efficiency
    - Branching point of end host A: a bifurcate router closest to A who has another face that can receive faster than A
    - There is a feedback loop between each end host and its branching point (see the branching point as the data provider)
    - Network friendliness *on each branch* is ensured (even when the bottleneck is not directly under the branching point)



Hybrid Solution

- Category 3: Network Assisted
  - Discussion on scalability of the solution
    - 1 subscription for each notification, similar to query/response
    - 1 count for a flow, much smaller than PIT (one entry for each data packet)
  - Further issues
    - Intermittent packets do not indicate congestion any more
      - How to do AIMD on the end host?
    - Fast receivers will receive all the packets and cause a huge window
      - How to limit that?

Reliability

- Receivers would lose packets in multi-rated solution, network-assisted solution and pure network-based solution
- Cache and other end hosts can help retrieving the missing packets
  - Names / Hashes for nameless contents
- Benefit of ICN
  - Privacy: The requester do not have to reveal its identity
  - Trust: Signed data to ensure integrity

Annex: Routing Notifications

- Consideration outside the core proposal.
- Two types of handling Routing Notifications Prefixes
  - **Stateless Forwarding (PUSH):** The notification prefixes are treated like other routing prefixes and shares the FIB used by the PULL traffic. These prefixes can also be marked in the routing control plane to provide logical separation while processing Notification flows. E.g. when the PUB/SUB service maintains
  - **Statefull Forwarding (PUSH)**: PUSH forwarding state is managed separately from the FIB state. Hence new control/service plane can be introduced to manage this state.

Annex : Notification Reliability

- Notification Reliability: The proposed PUSH is a best-effort network service. Applications manage reliability
  - Caching : Though this proposal doesn’t suggest any kind of caching, it can be explored to improve reliability. E.g. consumers can use PULL to recover lost notifications from nearby caches.
    - In a satefull forwarding situation, networks can proactively PUSH cached data, as subscriber state information can be associated with forwarding plane.
  - Notification Acknowledgement: ACK can be used at the application level to improve reliability.
    - In stateful forwarding situation, wherein ACK mechanisms can be investigated to be applied in the network layer with the help of caching.

Annex: Use Case Scenario

- **Towards PUB/SUB System** :
  - Used for granular topic based PUSH service.
  - Designed to meet any requirements : low latency, reliability, fast recovery, scalability, security, minimizing false (positive/negative) notifications.
  - CCN based PUB/SUB systems can leverage its multicast (proposed Notification Primitve), name based state, caching (needs exploration), pub/sub state (Statefull forwarding) to allow more efficiency, flexibility, reliability in meeting PUB/SUB objectives.
  - Two Design Approaches
    - Randezvous Approach : Centralized Service maps subscribers topics to published content.
    - Distributed Approach : This leverages the statefull forwarding approach. Where subscription table subscription state maintained in the forwarder, e.g. COPSS [1]. COPSS manages a subscrition Interest table, the content's content descriptors are matched to these entries to push Notifications to its downstream routers.

[1] Chen, J., Arumaithurai, M., Jiao, L., Fu, X., and K. Ramakrishnan, "COPS: An Efficient Contention Oriented Publish/Subscribe System.", ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS 2011) , 2011.

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.
- Discussions on flow and congestion control has been supplemented.

Next Steps

- Comments

Backup