1-to-n Matching between Interest and Content Objects for Reduction of Router Workload

Jun Kurihara

KDDI R&D Labs., Japan

ICNRG @ IETF 94
Yokohama, Japan, Nov. 5, 2015

\(^1\)The material was originally presented at IEEE CCN 2015 [KYUT15].
To propose a new research item on the CCN message relationship that should be considered in the community.
Fundamental rule of CCN

An interest $\iff$ A content object

One-to-one matching

Should this be always guaranteed?
Router’s processing of incoming messages

For each incoming message, search operations are needed at FIB/CS/PIT.
The router workload to search CS/PIT/FIB for incoming interests is likely to be serious in such a case.
Motivation

By aggregating multiple (mutually-related) interests into one request, the search complexity can be dramatically reduced.

Content objects

1-to-n matching

Corresponding interests
We introduced the *list interest* in IEEE CCN 2015

A new message that realizes the *light-weight* processing of requests for large content by co-operating the manifest in CCN 1.0.

This is an instance realizing the 1-to-*n* matching in CCN 1.0.
Related studies on aggregation of interests

- [BLJ13]: Specifying the “range of chunk numbers” in one interest to request multiple content objects.

⇒ Aggregates interests with the common name prefix, and enables *to skip most of FIB search*.

⇒ This doesn’t support
  - hash-based validation of content objects at intermediate routers,
  - matching with nameless objects (in CCNx1.0) at routers due to the lack of hash restrictions in interests
1 Introduction

2 Design of list interests

3 How much workload can be reduced?

4 Consideration on the deployment

5 Conclusion
Introduction

Design of list interests

How much workload can be reduced?

Consideration on the deployment

Conclusion
A manifest is a type of content object introduced in CCN 1.0.

- Manifest gives enumerated lists of content objects constituting a content.
- Each content object is specified by \((\text{ChunkNumber}, \text{Hash})\) pair and content name prefix.

A user first retrieve the manifest to obtain the content object list for the content.
Observation:

- A user obtains the content object list via manifest.
- The name prefix is common to all content objects in the list.

⇒ Interests for ones in the list must be routed to the same destination.

⇒ FIB search at a router must give the same result for all of them.
Key idea from this observation

We can skip most of FIB searches by aggregating the requests for content objects specified in the list.

NOTE: FIB search cost can be larger than CS/PIT search costs due to the search of longest-matching-prefix.
How to create a list interest from a manifest

List interest: A container of multiple \((\text{Chunk}#, \text{Hash})\) pairs

The user who received a manifest create the list interest just by copying the list in the manifest to the header.
The name of list interest itself has to be given in such a way that this can be routed to the correct destination.
How to process list interests at routers

It can be viewed as a simple parallelization of standard processing.

- CS/PIT search \( \Rightarrow \) Same times as standard interests for listed (Chunk#,Hash)'s.
- FIB search \( \Rightarrow \) Just once for the list interest itself.
- The list is updated after CS/PIT search for all contained pairs.
1 Introduction

2 Design of list interests

3 How much workload can be reduced?

4 Consideration on the deployment

5 Conclusion
Fix the router to process the list interest or individual interests. Fix the set of interests and corresponding Name-Hash pairs.

**List size** $L$: # of contained (Chunk#, Hash) pairs

$C_{\text{List}}$: router’s processing complexity for the list interest of size $L$

$C_{\text{Individual}}$: router’s processing complexity for the standard $L$ interests

$L_1$: # of cache-hits in $L$ interests/pairs

$L_2$: # of PIT-hits in $L - L_1$ interests/pairs

$L \geq L_1 + L_2$
\[ \frac{C_{\text{List}}}{C_{\text{individual}}} \approx \frac{LC_{\text{SearchCS}}}{LC_{\text{SearchCS}}} + \frac{(L - L_1)C_{\text{SearchPIT}}}{(L - L_1)C_{\text{SearchPIT}}} + \frac{C_{\text{SearchFIB}}}{(L - L_1 - L_2)C_{\text{SearchFIB}}} \]

\[ \Rightarrow \text{Difference} = \text{The number of FIB look-ups} \]
Comparison of the router workload

[Assumptions] $3.08 C_{\text{SearchCS}} = C_{\text{SearchFIB}}^2$, no cache-hit and no PIT-hit ($L_1 = L_2 = 0$)

The ratio of the complexities of $C_{\text{List}}/C_{\text{Individual}}$ for $L$

$\Rightarrow C_{\text{List}}$ is at most approximately 40% of $C_{\text{Individual}}$

$\Rightarrow C_{\text{List}}/C_{\text{Individual}} < 1$ even for $L = 2$

$3.08$ is the minimum # of look-ups for a hash table to find the longest-prefix-match in FIB [SNO13]
Thus we can see...

By introducing list interests, the router workload can be dramatically reduced from the standard interest-based request.
1 Introduction

2 Design of list interests

3 How much workload can be reduced?

4 Consideration on the deployment

5 Conclusion
Observation from the preliminary estimation

This shows that as $L$ increases, the router workload decreases.

But, we need a congestion control designed for the list interest (for $L$) to control the number of responses to issued list interests.
AIMD-based congestion control for list interest

$W$: Window size for list interests

$P$: # of in-flight content object

1: initialization: $W \leftarrow L, P \leftarrow L$
2: if Receive content object until RTO then
3: if Is slow start phase then
4: $W \leftarrow W + 1$
5: else (congestion avoidance phase)
6: $W \leftarrow W + 1/W$
7: end if
8: else
9: $W \leftarrow \max\{W/2, L\}$
10: end if
11: $P \leftarrow P + 1$
12: while $W \geq P + L$ do
13: Pack $L$ interests into a list interest and send it
14: $P \leftarrow P + L$
15: end while

TCP-like window control

New list interest is generated when $W-P > L$ (Waiting for the sufficient size of window)

Simple extension of AIMD-based congestion control [SGB12]
This algorithm did not harm the throughput of content retrieval for any $L$ in our simple simulation.

(The maximum possible $L = 25$ due to MTU=1280)

Simulation result

<table>
<thead>
<tr>
<th>List size $L$</th>
<th>1</th>
<th>10</th>
<th>20</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ave. throughput (Mbps)</td>
<td>9.59</td>
<td>9.61</td>
<td>9.61</td>
<td>9.61</td>
</tr>
</tbody>
</table>
Introduction

Design of list interests

How much workload can be reduced?

Consideration on the deployment

Conclusion
Conclusion

We proposed a new research item on CCN: 1-to-$n$ matching between interest and content objects

- List interest is one instance to realize such 1-to-$n$ matching in CCN 1.0 for reduction of router workload.

Potential research items on the 1-to-$n$ matching

- Congestion control strategy for 1-to-$n$ matching (end-to-end/hop-by-hop)
- More flexible PIT/CS structures for aggregated interests.
- etc.

