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]

[1] The material was originally presented at IEEE CCN 2015 [KYUT15].

Purpose of my talk

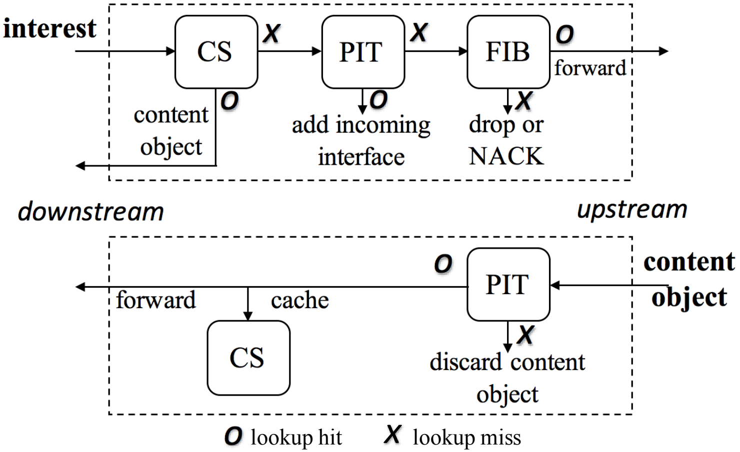
To propose a new research item on the CCN message relationship that should be considered in the community.

Fundamental rule of CCN

$$\underbrace{\text{An interest} \Longleftrightarrow \text{A content object}}_{\text{One-to-one matching}}$$

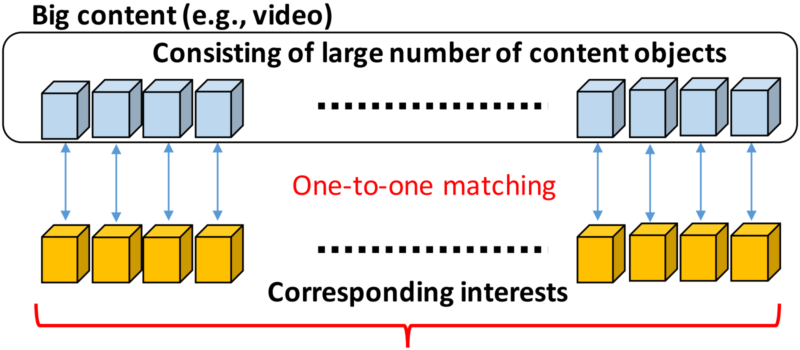
Should this be always guaranteed?

Router's processing of incoming messages



For each incoming message, search operations are needed at FIB/CS/PIT.

Motivation

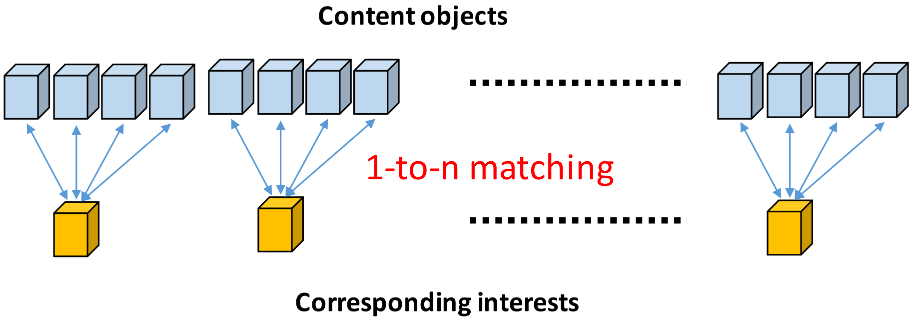


Large number of interests has to be issued to obtain the big content

⇒ 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.



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.

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

$\Rightarrow$ 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

Agenda (of IEEE CCN 2015 presentation)

1. Introduction
2. Design of list interests
3. How much workload can be reduced?
4. Consideration on the deployment
5. Conclusion

Background of the list interest design

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

List for a named content /parc/obj

Content name prefix	ChunkNumber	Hash
/parc/obj/	1	0xABCD
	2	0x1234
	3	0xA1B2
	4	0xC3D4
	...	...

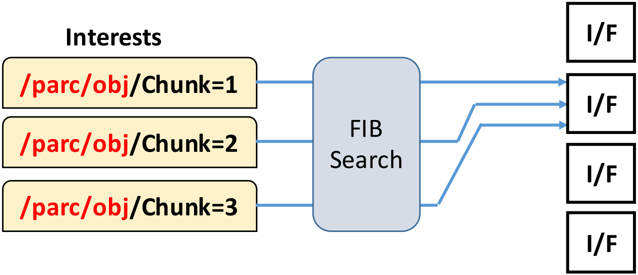
Manifest

- Manifest gives enumerated lists of content objects constituting a content.
- Each content object is specified by `(ChunkNumber,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.

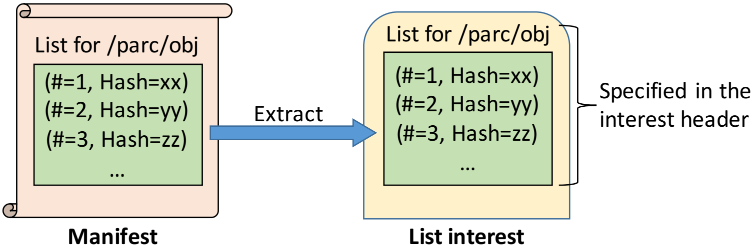
Key idea from this observation

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.

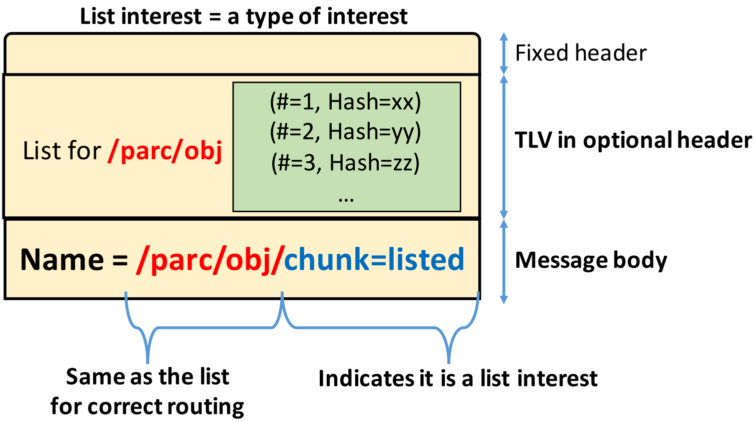
How to create a list interest from a manifest

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



The user who received a manifest create the list interest just by copying the list in the manifest to the header.

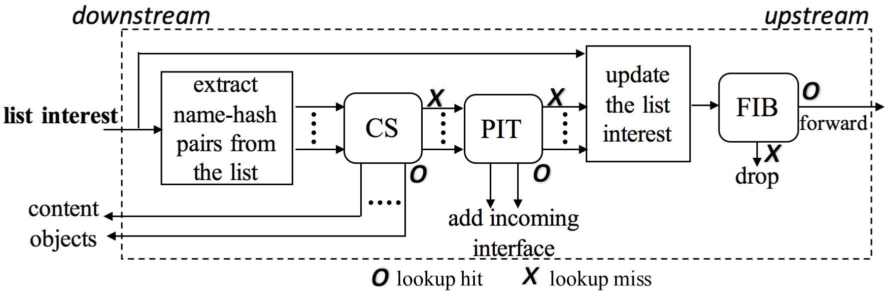
Design of list interests



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

Preliminary estimation of the router workload

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_{List}: router's processing complexity for the list interest of size L
$C_{\text{Individual}}$: router's processing complexity for the standard L interests

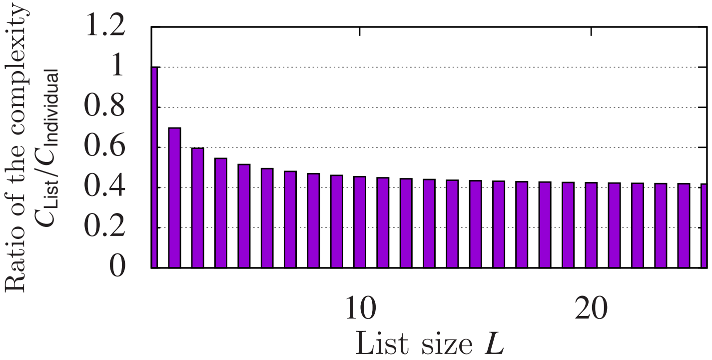
L_1: # of cache-hits in L interests/pairs
L_2: # of PIT-hits in $L - L_1$ interests/pairs
$L \geq L_1 + L_2$

$$C_{\text{List}}/C_{\text{individual}}$$
$$\simeq \frac{LC_{\text{SearchCS}} + (L - L_1)C_{\text{SearchPIT}} + C_{\text{SearchFIB}}}{LC_{\text{SearchCS}} + (L - L_1)C_{\text{SearchPIT}} + (L - L_1 - L_2)C_{\text{SearchFIB}}}$$

$\Rightarrow$ Difference = 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

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

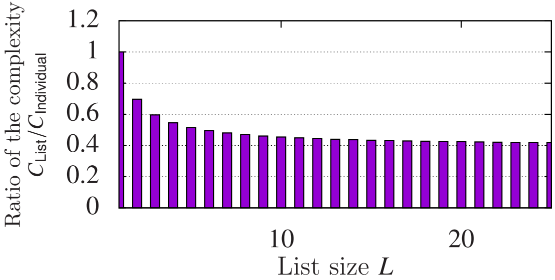
⇒ $C_{\text{List}}/C_{\text{Individual}} < 1$ even for $L = 2$

[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.

Observation from the preliminary estimation



$\Downarrow$

This shows that <u>as L increases, the router workload decreases.</u>

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

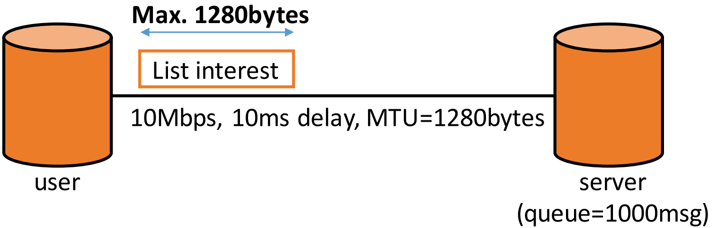
```
initialization: W ← L, P ← L
if Receive content object until RTO then
    if Is slow start phase then
        W ← W + 1
    else (congestion avoidance phase)
        W ← W + 1/W
    end if
else
    W ← max{W/2, L}
end if
P ← P + 1
while W ≥ P + L do
    Pack L interests into a list interest and send it
    P ← P + L
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

List size L	1	10	20	25
Ave. throughput (Mbps)	**9.59**	**9.61**	**9.61**	**9.61**

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.

References

[BLJ13] D. Byun, B.-J. Lee, and M.-W. Jang, "Adaptive flow control via interest aggregation in CCN," in *Proc. IEEE ICC 2013*, Jun. 2013, pp. 3738–3742.

[KYUT15] J. Kurihara, K. Yokota, K. Ueda, and A. Tagami, "List interest: packing interests for reduction of router workload in ccn 1.0," in *Proc. IEEE CCN 2015*, Dallas, TX, USA, Oct. 2015.

[SGB12] D. Saucez, L. A. Grieco, and C. Barakat, "AIMD and CCN: past and novel acronyms working together in the future Internet," in *Proc. ACM CSWS 2012*, 2012, pp. 21–26.

[SNO13] W. So, A. Narayanan, and D. Oran, "Named data networking on a router: fast and DoS-resistant forwarding with hash tables," in *Proc. IEEE/ACM ANCS 2013*, Oct. 2013, pp. 215–225.