NFD - Feature #1282

Design duplicate suppression on multicast face

02/23/2014 09:28 AM - Junxiao Shi

Status:	Feedback	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	20%
Category:	Faces	**Estimated time:**	6.00 hours
Target version:			

Description

Design a Data packet duplicate suppression mechanism on multicast face.

ccnd's algorithm is:

- Multicast datagram face should delay every outgoing Data packet for a random duration (on the order of 10ms).
- During the delay of a packet, if an identical packet is received on the face, cancel the send.

but it causes packet reordering and cannot adapt to link condition.

This Task is to design a better algorithm for duplicate suppression.

Category is currently "Faces", but the design MAY propose to put part of this feature in Forwarding/Tables if necessary.

Related issues:

Follows NFD - Task #1189: UdpFace implementation	**Closed**

History

#1 - 02/23/2014 09:31 AM - Junxiao Shi

20140220 conference call decides:

- Duplicate suppression applies to Data packets only, because delaying Interests causes too much end-to-end delay.
- Duplicate suppression with random delay will cause packet reorder. Congestion control scheme should not expect in-order delivery.

#2 - 02/26/2014 09:23 AM - Davide Pesavento

1/ I think this can be postponed until after the first release... it's just an optimization anyway.

2/ Duplicate suppression applies to multicast Ethernet faces too, so it should be factored out in a common implementation, if possible.

#3 - 02/27/2014 08:37 AM - Junxiao Shi

- Target version changed from v0.1 to v0.2

20140227 conference call agrees to this MAY be deferred to Version 2.

#4 - 05/14/2014 10:19 AM - Junxiao Shi

- Assignee deleted (Giulio Grassi)

- Target version deleted (v0.2)

20140514 call decides:

ccnd's algorithm

- causes packet reordering, which is not desired
- has constant maximum delay duration; it may need to be adaptive to link conditions

We should evaluate and design the algorithm before implementation.

#5 - 11/14/2014 01:31 PM - Junxiao Shi

- Assignee set to Junxiao Shi

- Target version set to v0.3

20141114 conference call decides that Junxiao could start thinking about the design.

Implementation will be a separate Task after design is approved.

#6 - 02/07/2015 06:18 AM - Junxiao Shi

- Subject changed from Duplicate suppression on multicast face to Design duplicate suppression on multicast face

- Description updated

- Status changed from New to In Progress

- Estimated time changed from 5.00 h to 3.00 h

I have some ideas and will make slides.

#7 - 02/11/2015 10:15 AM - Junxiao Shi

- File mcast-dup-suppress_20150211.pptx added

- % Done changed from 0 to 10

- Estimated time changed from 3.00 h to 6.00 h

I have read ccnd code around this area. Attached mcast-dup-suppress_20150211.pptx is my understanding.

I don't understand why ccnd algorithm would cause packet reordering.

My initial idea turns out to be somewhat similar to what's already in ccnd.
I need to understand the cause of packet reordering before continuing with this design.

#8 - 02/14/2015 08:05 AM - Junxiao Shi

- File mcast-dup-suppress_20150214.pptx added

- Status changed from In Progress to Feedback

- Assignee deleted (Junxiao Shi)

- % Done changed from 10 to 20

20150211 meeting with Beichuan clarifies that ccnd's packet reordering is observed in an experiment with an unspecified "old version" of ccnd (but I don't see any change in the duplicate suppression algorithm since ccnd v0.1.0), and there's no evidence for ccnd v0.7.2 algorithm to have packet reordering problem.

I feel that duplication suppression is not the most effective way to solve the "return Data packets from multiple upstreams waste bandwidth" problem, because all upstreams must still producer/fetch the Data in response to every Interest, although all but one upstream's work will be wasted. A better solution is multicast/unicast switch. Duplication suppression will still be useful after that's available, but is less important.

Attached slides show the ideas of both duplication suppression and multicast/unicast switch, and their comparison.

Given its ineffectiveness, I don't plan to further work on duplicate suppression design recently.

#9 - 02/18/2015 02:03 PM - Junxiao Shi

- Target version deleted (v0.3)

20150218 conference call agrees with note-8.

"Target version" is cleared. This issue can restart after multicast/unicast switch is designed.

#10 - 04/18/2015 12:47 PM - Junxiao Shi

- Status changed from Feedback to New

#11 - 09/16/2018 06:57 AM - Ernest McCracken

- Assignee set to Ernest McCracken

Hi I have started working on this data reply suppression based on random delay wait. Our implementation was specific to multi access only. I will read up on previous work and update accordingly.

#12 - 09/24/2018 09:05 PM - Ernest McCracken

Some notes after reviewing the slides and my own implementation of data cache reply suppression.

- We found that data reply suppression can be accomplished solely in a forwarding strategy. However, a switch seems like it would require a naming schema.
- Link conditions for data reply suppression could also be tracked within the forwarding strategy via measurements table.
- Multicast/Unicast switch guaranties a single data reply. Data reply suppression does not.

- More than one reply may be beneficial in high loss link conditions.
- Our data cache suppression works on content store hits. Therefore it works for upstream NFD's that have the data cached but are not the original data providers.
- In response to "all upstreams must still producer/fetch the Data in response to every Interest, although all but one upstream's work will be wasted." for a multicast/unicast switch there is an additional network overhead of responding to the initial interest where in data reply suppression the overhead is a randomized delay. If link conditions can be monitored the data reply suppression delay could result in less overall delay than two rounds of interests being sent.
- Could a multicast/unicast switch be implemented at an application library level? This would avoid altering NFD code base.

#13 - 09/28/2018 01:35 AM - Ernest McCracken

- Status changed from New to In Progress

Another issue with a multicast/unicast switch is the multicast group will have a decreased caching efficiency due to data packets not being forwarded back to the multicast face but instead using the unicast return tunnel back to the sender. Multicast groups especially on broadcast mediums would lose cache efficiency significantly.

If we are able to monitor RTTs for multicast faces we can create a random delay based on average RTT plus the additional accepted delay of the data suppression algorithm. This way the delay responds to changes in link conditions.

I think with the caching and the simplicity in its implementation(a forwarding strategy) data reply suppression is the best route.

A multicast/unicast switch still seems it would be useful especially in the multicastStrategy class where the multicasting is name based.

#14 - 05/02/2019 08:42 AM - Ernest McCracken

- Assignee deleted (Ernest McCracken)

Removing myself from this for the time being since I have not had any time to work on the multicast/unicast switch itself.

#15 - 05/06/2019 02:56 PM - Ernest McCracken

- Assignee set to Ernest McCracken

#16 - 04/09/2020 01:32 PM - Davide Pesavento

- Tracker changed from Task to Feature

- Status changed from In Progress to Feedback

- Assignee deleted (Ernest McCracken)

#17 - 04/09/2020 01:36 PM - Saurab Dulal

- File Interest and Data Reply Suppression.pptx added

#18 - 04/09/2020 02:09 PM - Junxiao Shi

2020-04-09 NFD call discussed this issue.

Davide claims that you **cannot suppress both Interest and Data** due to hidden terminal problem.
In the topology below, A and B can hear each other, and B,C,D can hear each other.

1. Both A and C send an Interest. B hears both Interests.
2. D replies with Data. B hears the Data.
3. If B suppresses Data forwarding, A would not be able to retrieve the Data.

```
A    B    C
          D
```

Lan explains that Alvy has a solution to the hidden terminal problem: suppression probability.
By tuning the probability, it is possible to adjust how many duplicates you get, without breaking communication above.

Lixia suggests that we can **start with one-hop WiFi ad hoc scenario**.
However, it is unclear who is using NDN WiFi ad hoc, other than in simulations and completely fabricated research scenarios. In particular, Android does not support WiFi ad hoc at all.
Lan mentions vehicular use case (DSRC), but Davide believes it's no longer relevant in industry.
Lixia believes that industry's ignorance on ad hoc is because the lack of mature solution.

Alex mentions WiFi mesh that use peer-to-peer communication.
It allows WiFi to have *mostly* point-to-point communication without access point, but the protocols support multicast as well.
Davide explains that it uses HWMP protocol to establish a routing table, but this solution can only work with mostly static network and will break down if there's too-fast mobility.

Alex also mentions sensor networks including Zigbee that are fundamentally ad hoc.
No further discussion occurred on this scenario.

It's still being debated **whether this feature is part of face or part of forwarding/strategy**.
Lixia insists that this feature should belong in the face, because this is a L2 feature and suppression should occur per face.
Both Alvy and Alex have experimental implementation that have this feature in the strategy, because the strategy is already remembering relevant information in the tables.
No agreement was reached in this point.

#19 - 04/09/2020 02:44 PM - Davide Pesavento

Junxiao Shi wrote:

> Davide claims that you **cannot suppress both Interest and Data** due to hidden terminal problem.

It's actually worse than that. The scenario in the example breaks down even if you *only* have *Data* transmission suppression. And you have a similar problem with *only Interest* suppression.
(also note that "hidden terminal" is a term I made up, it is not related to the hidden terminal problem in 802.11)

> Lan explains that Alvy has a solution to the hidden terminal problem: suppression probability.
> By tuning the probability, it is possible to adjust how many duplicates you get, without breaking communication above.

Yes but I'm not sure if I'd call that a "solution", it's not clear to me how to effectively set this probability. If it's too low, some nodes may need several retransmissions before obtaining the content; if it's too high, then you still have a large number of duplicate Data packets in the network.
Maybe Data suppression should be disabled (or probability of suppressing should be greatly reduced) if a node receives the same Interest more than once in a short period of time?

Files

mcast-dup-suppress_20150211.pptx	42.2 KB	02/11/2015	Junxiao Shi
mcast-dup-suppress_20150214.pptx	48 KB	02/14/2015	Junxiao Shi
Interest and Data Reply Suppression.pptx	1.45 MB	04/09/2020	Saurab Dulal