

NFD - Bug #3411

NccStrategy doesn't explore potential upstreams

01/14/2016 01:09 PM - Hila Ben Abraham

Status:	Closed	Start date:	01/14/2016
Priority:	Normal	Due date:	
Assignee:	Hila Ben Abraham	% Done:	100%
Category:	Forwarding	Estimated time:	3.00 hours
Target version:	v0.5		
Description			
Environment:			
A---B C---D			
<ul style="list-style-type: none">• consumer on A, producer on D• A-B-D link RTT is 20ms• A-C-D link RTT is 20ms			
Steps to reproduce:			
<ol style="list-style-type: none">1. send Interests from A to D2. observe packets sent through B & C.3. wait until the exploration period is over (ncc selected a best face and adjusted the prediction to be higher than the path RTT).4. reduce the RTT of the non-selected path to 10 ms. (If B was selected as best next hop, change A-C-D link RTT to 10ms).			
Expected: ncc adjusts the prediction down until best face times out, and then explores and selects the other face.			
Actual: ncc doesn't explore the second face and continue to use the previously selected face as the best face.			
I believe this bug is caused by undesired call of timeoutOnBestFace. Unlike doPropagate, timeoutOnBestFace is not canceled when the interest is satisfied, but only when PITEntryInfo is deleted. I suspect that in some cases, timeoutOnBestFace is scheduled to run before the pit entry is erased, and therefore the prediction is adjusted up even though the interest was satisfied. (The prediction is adjusted down upon the arrival of the Data packet and then immediately adjusted up).			
This bug can cause undesired behavior in other scenarios as well.			

History

#1 - 01/14/2016 01:17 PM - Hila Ben Abraham

- File *nccPrediction.jpg* added

Attached graph shows ncc prediction over time in the described scenario. As shown, the prediction doesn't go below the link RTT and therefore the strategy does not explore additional faces.

#2 - 01/15/2016 01:36 PM - Hila Ben Abraham

- File *nccPrediction.jpg* added

The updated graph shows ncc v1, which is a modified version of ncc. The only difference between the modified v1 and the existing one is that I cancel the scheduled call for timeoutOnBestFace if the interest is satisfied by the best face (inside beforeSatisfyInterest).

You can see that the prediction is now around the expected actual data response time (link RTT of 20ms + 8ms of App response time).

However, ncc still doesn't always explore additional faces due to the following sequence of events:

1. Prediction is smaller than the link RTT ==> best face expires. timeoutOnBestFace is triggered and prediction is adjusted up.
2. Data is received on the best face. beforeSatisfyInterest is triggered and cancel the scheduled call of doPropagate.

#3 - 01/15/2016 01:52 PM - Junxiao Shi

- Assignee set to Hila Ben Abraham

- Target version set to v0.5
- Estimated time set to 3.00 h

I compared NccStrategy with ccnd 0.7.2 and agree this is a bug.

- ccnd 0.7.2 cancels the timer in finalize_interest, which is called when a PIT entry is erased, which in turn happens as soon as the Interest is satisfied.
- In NFD, when the best face answers, the Interest table entry is retained and is scheduled to be erased after "straggler timer". This breaks the assumption of NccStrategy.

I agree with the proposed solution in note-2.

Hila, can you upload a Change along with a test case?

#4 - 01/15/2016 02:02 PM - Junxiao Shi

However, ncc still doesn't always explore additional faces due to the following sequence of events:

1. Prediction is smaller than the link RTT ==> best face expires. timeoutOnBestFace is triggered and prediction is adjusted up.
2. Data is received on the best face. beforeSatisfyInterest is triggered and cancel the scheduled call of doPropagate.

This is a weakness in ccnd's design.

Under ccnd strategy, a new nexthop can become default only if (RTT of new nexthop + current 'prediction' < RTT of current nexthop). Since 'prediction' is only slightly lower than RTT of current nexthop ($1/2^7$), this condition is unlikely to be met.

An experiment with ccnd 0.7.2 exhibits a similar behavior: strategy is stuck with slower path even if a faster path has recovered.

Since NccStrategy is designed to match ccnd 0.7.2 behavior, this part should not be changed.

#5 - 01/16/2016 09:04 AM - Hila Ben Abraham

Hila, can you upload a Change along with a test case?

yes.

Since NccStrategy is designed to match ccnd 0.7.2 behavior, this part should not be changed.

Is there a plan to improve ncc as a new nfd strategy?

#6 - 01/16/2016 09:56 AM - Junxiao Shi

Is there a plan to improve ncc as a new nfd strategy?

No. There isn't sufficient understanding of ccnd strategy's use case, design rationale, and behavior.

#7 - 01/20/2016 12:59 PM - Hila Ben Abraham

- Status changed from New to In Progress

#8 - 01/20/2016 01:13 PM - Hila Ben Abraham

Should the test case check the correctness of 'prediction' after data received on time? or should it just validate the timeout cancellation? (or maybe both?) Are there any special instructions/resources regarding the development of a new test case?

#9 - 02/07/2016 07:30 AM - Junxiao Shi

Answer to note-8:

No, don't test the *minute detail* in the strategy, such as the exact timeout values.

Instead, test the *trend*, such as whether it's increasing/decreasing when expected.

See [UnitTesting](#) page on test suite guidelines.

#10 - 03/07/2016 11:40 PM - Alex Afanasyev

- Status changed from In Progress to Code review

- % Done changed from 0 to 100

#11 - 03/10/2016 05:40 AM - Junxiao Shi

I've written a test case Fw/TestNccStrategy/PredictionAdjustment in <http://gerrit.named-data.net/2736> patchset7.

Without Hila's changes in NccStrategy, the prediction would increase all the way to the maximum.

With the changes in NccStrategy, the prediction is able to converge near path RTT.

Thus, I believe the fix is correct.

The same test case also checks whether the new best face would be selected.

This is expected to fail due to the weakness described in note-4.

I'm using a BOOST_WARN to generate a warning only.

#12 - 03/10/2016 03:09 PM - Alex Afanasyev

Make it expected failure. Otherwise, there is no point of doing the check.

#13 - 03/11/2016 10:44 AM - Hila Ben Abraham

I agree that patchset 7 includes a more comprehensive test scenario for ncc than the previous patchsets (that were mainly focused on this bug fix).

The same test case also checks whether the new best face would be selected.

This is expected to fail due to the weakness described in note-4.

I'm using a BOOST_WARN to generate a warning only.

I think that this test case should not be included in the test scenario. While the original ncc design does have the described weakness, my recent research efforts show that the current implementation of ncc in NDN actually requires that current 'prediction' + RTT of new next hop/2 (one way) < RTT of current next hop/2 to switch to the better performing face. This difference is caused by to the NACK generated on node D. I can elaborate more on that if needed, but I just want to point it out here due to the explanation of the expected failure in note-4 and on gerrit.

#14 - 03/11/2016 07:29 PM - Junxiao Shi

I disagree with note-13.

The second part of my test case shows the desired behavior as described in the Description field on this issue, but it's unavailable due to a weakness in ccnd design, and therefore should be marked as an expected failure.

#15 - 03/14/2016 09:59 AM - Hila Ben Abraham

I am not saying there is no weakness in the ccnd design, but I don't think note-4 (cited in patchset 7) captures the entire picture. In the current implementation of NDN the ncc strategy won't explore potential upstreams even if RTT of new nexthop + current 'prediction' > RTT of current nexthop. So this is more than just design weakness.

Anyway, if you still want to keep it I agree with note-12.

#16 - 03/15/2016 10:04 AM - Junxiao Shi

The PredictionAdjustment test case in patchset7 indicates that "switching to new best face" did not occur under the delay values set forth in the test case, due a weakness in [CcndStrategy](#) design.

This does not reflect the whole picture (last paragraph of note-13), but it's a fact of the design, so I intend to keep this part of the test case.

Can you update it according to note-12?

#17 - 03/31/2016 01:03 PM - Junxiao Shi

- Status changed from Code review to Closed

Files

nccPrediction.jpg	26 KB	01/14/2016	Hila Ben Abraham
nccPrediction.jpg	27.4 KB	01/15/2016	Hila Ben Abraham