NFD - Feature #3566
Adaptive Forwarding Strategy for hyperbolic routing

03/23/2016 12:27 PM - Junxiao Shi

Status: Closed
Priority: Normal
Assignee: Vince Lehman
Category: Forwarding
Target version: v0.5
Start date:
Due date:
% Done: 100%
Estimated time: 18.00 hours

Description
Current forwarding strategies in NFD focus on being able to retrieve content, but may not always find the lowest delay path. To facilitate optimal forwarding in HR and address drawbacks (embedding issues, network dynamics, etc.), a strategy which bases its forwarding decisions on RTT measurements and also periodically probes alternative next hops is crucial to HR’s deployment.

The design of the adaptive forwarding strategy is composed of two components:

**SRTT-based Forwarding**

When a Data packet is received, the RTT is recorded (time between Interest forwarded and Data received). This RTT is used to compute an SRTT; an SRTT measurement is calculated and associated with the FIB entry which the Interest matched.

When an interest arrives, all available next hops from the FibEntry are divided into three distinct groups:

- Group 1: Next hops with SRTT values
- Group 2: Next hops that do not have measurements
- Group 3: Next hops that timed-out on the previous Interest

The strategy will then try to pick a next hop in the following way:

1. If Group 1 is not empty, pick the next hop with the lowest SRTT
2. Otherwise, if Group 2 is not empty, pick the next hop with the lowest cost
3. Otherwise, pick the lowest cost next hop from Group 3

**SRTT-based Probing**

The first probe for a namespace is scheduled after the first Interest which matches the FIB entry arrives. The first probe is scheduled at a random interval from [0, 5] seconds. After the initial probe, probing is scheduled to occur periodically every 60 seconds.

If probing is due but no Interests arrive under that namespace, probing will not occur until a matching Interest arrives. Probes use the same name as the triggering Interest but use a different nonce. A different nonce is used to ensure that if the probed path is able to reach the Data, the Data will be returned on the probed path as well as the primary path.

The steps for probing are defined as follows:

1. Pick the next hop with the lowest routing cost from Group 2
2. Otherwise, sort the next hops in Group 1 by SRTT and Group 3 by cost; priority Group 1 > Group 3
3. Assign a probing probability to each next hop and probabilistically select a next hop for probing

The probing probability of the next hop with rank i = 1, 2, …, N in the sorted list is assigned P(i) = (2 * (N + 1 - i))/(N * (N + 1))

**Handling Incoming NACKs**

All received NACKs are handled by setting the timeout flag in the corresponding face’s measurements
Outgoing NACKs

On an incoming Interest if the strategy cannot forward to any next hops, send a NO_ROUTE NACK.

Retransmissions

- Consumer retransmissions
  - Follow the same suppressions logic as implemented in best route strategy v4 (#1913)
  - Follow similar forwarding logic for retransmitted Interest as in best route strategy v4 (#1871); On retransmission, pick best next hop from sorted groups
- Router retransmissions - this strategy does not perform router retransmissions

Implementation

An experimental implementation of the above described strategy can be found at https://github.com/vslehman/NFD/tree/asf/daemon/fw/asf.

Related issues:

| Has duplicate NFD - Feature #3813: Forwarding strategy for using multiple int... | Duplicate | 10/16/2016 |
| Has duplicate NFD - Task #1901: Strategy for backbone router | Duplicate | |

History

#1 - 03/23/2016 10:08 PM - Vince Lehman
- Subject changed from Smart-Flooding strategy for hyperbolic routing to Adaptive Forwarding Strategy for hyperbolic routing
- Description updated

#2 - 03/28/2016 11:48 AM - Vince Lehman
- Description updated

#3 - 03/29/2016 09:54 AM - Vince Lehman
- Description updated

#4 - 03/29/2016 12:25 PM - Vince Lehman
- File hr-adaptive-forwarding-strategy.pdf added

#5 - 03/29/2016 09:13 PM - Klaus Schneider
Hi Vince,

sorry that I missed the NFD call today, I got confused by some change in the timezone difference (Tucson apparently doesn't have daylight saving time).

Some comments:

1. It looks like delay probing is incredibly cheap (one packet every 60s/10s). Wouldn't it thus be easier to just probe all faces (which are not the current primary face) every x seconds? This might save some coding and debugging effort.

2. As I understood, you are recording the RTT measurements of the regular data for the primary path and the RTT measurements of probes for the probed paths. Therefore, the primary path receives many more RTT measurements, and thus quicker updates, than the probed paths. This might lead to problems, such as the following:

Let's say you have the following topology. Router A has two faces to R1,R2, but they both join to the same link (C,P).

```
----- R1 ----- |
|           |
A --- C ----- P |
|           |
-----R2------ |
```

Let's say (A,R1) is the primary path. Now, when the delay at (C,P) increases, A will switch to R2 as primary path. After some time, A will notice that
the delay on R2 is actually higher, and switch back to R1.

A solution would for all paths (also the primary path) to only consider RTT measurements during probing time (probing all faces the same time improves the reliability here). However, it also slows down the time to detect delay changes at the primary path.

The scenario can become worse when the primary traffic of A actually causes the delay by increasing the queuing delay at the intermediate routers. In this case, A may indefinitely oscillate between R1, and R2.

In this case, you might want to consider some sort of hysteresis, that is, an additional threshold that encourages staying on the current face if the other faces are just slightly better.

I described the idea in my last paper (http://conferences2.sigcomm.org/acm-icn/2015/proceedings/p137-schneider.pdf) and have a prototype implementation here: https://github.com/schneiderklaus/ndnSIM/blob/master/NFD/daemon/fw/lowest-cost-strategy.cpp (although with a different scenario in mind).

1. Here is another problem when the secondary path overlaps with the primary path (as described above). You use a new nonce for the probes, which avoids the problem that the second interest will be dropped at router C. However, Router C will still put the second interest in the PIT-Entry and not send out a second data packet. Since the data packet is already on its way, the probing result of the slower path will be faster than its actual RTT.

Maybe this doesn't matter, because the interest should still be slower than the fastest path (the returning data still has a longer one-way delay). Another option would be to probe a name different than the current interest. Maybe with the a correct prefix, but unavailable (randomly generated) data name (getting a NACK back).

---

#6 - 03/30/2016 04:05 PM - Vince Lehman

Hi Klaus,

Thanks for the comments and questions.

Klaus Schneider wrote:

1. It looks like delay probing is incredibly cheap (one packet every 60s/10s). Wouldn't it thus be easier to just probe all faces (which are not the current primary face) every x seconds? This might save some coding and debugging effort.

I do want to clarify that although there is probing every 60s, this probing is per namespace. If there are many FIB entries with many next hops, trying every face for each probing period may not be very cheap (especially on highly connected nodes).

There are two main reasons we chose to probe only a single face at each probing period:

1. We want to take advantage of the routing metrics provided by hyperbolic routing to influence our interface selection for probing. It is possible that many of the interfaces are not useful for a particular name prefix; instead of probing these faces that will not lead to data, we try to limit our probes to interfaces we expect to be useful based on hyperbolic distance.

2. We want to keep the overhead constant regardless of the connectivity of the node; we want to constrain the overhead as topologies grow in size. By only probing one interface per namespace, the overhead is limited by the number of namespaces and the probing interval (i.e. 60 seconds).

Klaus Schneider wrote:
1. As I understood, you are recording the RTT measurements of the regular data for the primary path and the RTT measurements of probes for the probed paths. Therefore, the primary path receives many more RTT measurements, and thus quicker updates, than the probed paths. This might lead to problems, such as the following:

Let’s say you have the following topology. Router A has two faces to R1, R2, but they both join to the same link (C, P).

```
----- R1 -----
|          |
A          C ---- P
|          |
----- R2 -----
```

Let’s say (A, R1) is the primary path. Now, when the delay at (C, P) increases, A will switch to R2 as primary path. After some time, A will notice that the delay on R2 is actually higher, and switch back to R1.

A solution would for all paths (also the primary path) to only consider RTT measurements during probing time (probing all faces the same time improves the reliability here). However, it also slows down the time to detect delay changes at the primary path.

The scenario can become worse when the primary traffic of A actually causes the delay by increasing the queuing delay at the intermediate routers. In this case, A may indefinitely oscillate between R1, and R2.

In this case, you might want to consider some sort of hysteresis, that is, an additional threshold that encourages staying on the current face if the other faces are just slightly better.

I described the idea in my last paper (http://conferences2.sigcomm.org/acm-icn/2015/proceedings/p137-schneider.pdf) and have a prototype implementation here: https://github.com/schneiderklaus/ndnSIM/blob/master/NFD/daemon/fw/lowest-cost-strategy.cpp (although with a different scenario in mind).

Thanks for the suggestion, I will read over "Beyond Network Selection" again and look into this as it seems like it could be useful.

Klaus Schneider wrote:

1. Here is another problem when the secondary path overlaps with the primary path (as described above). You use a new nonce for the probes, which avoids the problem that the second interest will be dropped at router C. However, Router C will still put the second interest in the PIT-Entry and not send out a second data packet. Since the data packet is already on its way, the probing result of the slower path will be faster than its actual RTT.

Maybe this doesn't matter, because the interest should still be slower than the fastest path (the returning data still has a longer one-way delay). Another option would be to probe a name different than the current interest. Maybe with the a correct prefix, but unavailable (randomly generated) data name (getting a NACK back).

Yes, we think that since the Interest would be slower than the faster path and since we are using SRTT to smooth out these variations that this should not be a serious issue. If the strategy starts using the slower path due to caching on the path or cases like the above, eventually the SRTT of the slower path should become worse than the faster path and switch back.
Hi Klaus,

Thanks for the comments and questions.

Klaus Schneider wrote:

1. It looks like delay probing is incredibly cheap (one packet every 60s/10s). Wouldn't it thus be easier to just probe all faces (which are not the current primary face) every x seconds? This might save some coding and debugging effort.

I do want to clarify that although there is probing every 60s, this probing is per namespace. If there are many FIB entries with many next hops, trying every face for each probing period may not be very cheap (especially on highly connected nodes).

Okay, that makes sense. Are there any estimates of how many name prefix entries the average core router holds?

#8 - 03/31/2016 07:25 AM - Junxiao Shi

20160329 conference call approves the idea of this strategy.

It's pointed out that the development of adaptive forwarding strategy is independent from the adoption of hyperbolic routing. Hyperbolic routing may rely on adaptive forwarding strategy as a prerequisite, but developing adaptive forwarding strategy does not depend on the approval of hyperbolic routing deployment.

Before going into implementation, the design needs to be amended to clarify the following:

- How does the strategy handle consumer retransmissions?
- How does the strategy process incoming Nacks?
- Does the strategy generate outgoing Nacks?

I understand that the strategy was designed before these issues were raised, so now it's a good time to think above them. Although I recommend supporting all three features listed above, a design without these features will be acceptable to me, as long as the designer has thought about them.

The design may choose the adopt the same processing steps for consumer retransmissions and Nacks as another existing strategy, such as best-route v4. In that case, the design can simply reference the referenced strategy, but the implementation will need to copy over the relevant code (this is unfortunate, but will be improved after #2000).
Junxiao Shi wrote:

How does the strategy handle consumer retransmissions?
How does the strategy process incoming Nacks?
Does the strategy generate outgoing Nacks?

I've updated the description to include some answers to these questions. I am still working on the logic for handling DUPLICATE NACKs.

Klaus Schneider wrote:

As discussed in today's call:

The reaction to congestion NACKs probably requires agreement between the receiver and the sender of the NACK. Before implementing it here, we should specify when exactly the upstream sends a congestion NACK. Moreover, the reaction will likely be similar in many strategies and is a good candidate for task #2000.

Also, "Duplicate NACK" sounds like "Duplicate ACK" from TCP. Does it mean that you receive two NACKs for the same sequence number in a row? If not, I would suggest to find another name to avoid confusion.

Klaus Schneider wrote:

As discussed in today's call:

The reaction to congestion NACKs probably requires agreement between the receiver and the sender of the NACK. Before implementing it here, we should specify when exactly the upstream sends a congestion NACK. Moreover, the reaction will likely be similar in many strategies and is a good candidate for task #2000.

I agree; the congestion NACKs are specific to the congestion control implementation and so should be handled by an external module. This strategy will simply consider each NACK as a timeout.

Also, "Duplicate NACK" sounds like "Duplicate ACK" from TCP. Does it mean that you receive two NACKs for the same sequence number in a row? If not, I would suggest to find another name to avoid confusion.

Duplicate NACKs are sent when the forwarder receives an Interest with a duplicate nonce, i.e. on Interest loop detection.
After Receive Interest Trigger, should we send back a NACK if there is no faceToUse for forwarding or only reject the pending Interest?

Minsheng Zhang wrote:

After Receive Interest Trigger, should we send back a NACK if there is no faceToUse for forwarding or only reject the pending Interest?

Yes, I think a NACK should probably be sent in that scenario.

Hey Vince,

I just read the new specification and have a couple questions:

To facilitate optimal forwarding in hyperbolic routing (HR) and address drawbacks (embedding issues, network dynamics, etc.), a strategy must be able to make forwarding decisions based on Data retrieval delay

20160705 call decides the design of this strategy should be published as a technical report, and NFD devguide should cite this TR.
It is unclear to me why the strategy must be based on "data retrieval delay" rather than packet loss information or explicit signals? Maybe you can put in a longer rationale or a reference.

Group 3: Next hops that timed-out on the previous Interest

What do you consider as reasons for why the Interest timed out? What do you mean with "the previous Interest"? "Previous" according to the naming convention (/foo/N, /foo/N+1) or any interests sent out previously?

Let's say we are sending interests /foo/1, /foo/2, /foo/3, and so on. Let's also say Interest /foo/2 timed out and you detect the timeout just before you send out /foo/17. Do you then put the router into Group 3?

What if at this point /foo/3, /foo/4, and /foo/5 have been received successfully?

- What is the rationale for your two formulas of the probing probability?

Lastly, a more general question:

- Is saving measurement information (here: SRTT) per FIB entry fine-granular enough?

What if the your local FIB entry /foo splits up at later routers to /foo/A, /foo/B, /foo/C. Then you would be mixing RTT measurements for different prefixes and likely send some traffic on suboptimal paths.

This isn't specific to hyperbolic routing, so feel free to ignore. But it would still be interesting if there's a good solution already.

#23 - 07/11/2016 03:23 PM - Junxiao Shi
- Status changed from Code review to Feedback

Code is merged. NFD devguide needs a citation to the TR.

#24 - 07/23/2016 07:32 AM - Vince Lehman
I've pushed changes to the Developer's guide to add a description and citation for the ASF strategy.

#25 - 07/23/2016 02:04 PM - Junxiao Shi
- Status changed from Feedback to Closed
- % Done changed from 90 to 100
I fixed the error in nfd-docs:commit:db7f903c1f46864dd34d14a0aa5290b8efdead90 and also corrected strategy classes to be under namespace nfd::fw.

#26 - 10/28/2016 09:52 AM - Junxiao Shi
- Related to Feature #3813: Forwarding strategy for using multiple interfaces in generic deployments added

#27 - 10/28/2016 09:52 AM - Junxiao Shi
- Related to deleted (Feature #3813: Forwarding strategy for using multiple interfaces in generic deployments)

#28 - 10/28/2016 09:52 AM - Junxiao Shi
- Has duplicate Feature #3813: Forwarding strategy for using multiple interfaces in generic deployments added

#29 - 07/07/2017 01:36 AM - Junxiao Shi
- Has duplicate Task #1901: Strategy for backbone router added

Files

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>hr-adaptive-forwarding-strategy.pdf</td>
<td>569 KB</td>
<td>04/08/2016</td>
<td>Vince Lehman</td>
</tr>
<tr>
<td>asf-specification.pdf</td>
<td>199 KB</td>
<td>07/01/2016</td>
<td>Vince Lehman</td>
</tr>
</tbody>
</table>