Poster: In-Network Retransmissions in Named Data Networking

Hila Ben Abraham
Computer Science and Engineering
Washington University in St. Louis
hila@wustl.edu

Patrick Crowley
Computer Science and Engineering
Washington University in St. Louis
pcrowley@wustl.edu

ABSTRACT

A core mechanism in every forwarding strategy is the decision of whether to retransmit an unsatisfied Interest or to wait for an application retransmission. While some applications request control of all retransmissions, others rely on the assumption that the strategy will retransmit an Interest when it is not satisfied. Although an application can select the forwarding strategy used in the local host, it cannot guarantee that the same strategy will be selected in other nodes in the network, especially in shared resource environments. In this paper we propose a simple forwarding strategy abstraction that allows the application to decide whether a network retransmission is required, and to differentiate application retransmissions from network retransmissions.

1. INTRODUCTION AND MOTIVATION

The current design of Named Data Networking (NDN) [1] and Content-Centric Networking (CCN) [2] software prototypes permits the application developer to pair a forwarding strategy with its application namespace. One core mechanism of a forwarding strategy chooses what to do when an Interest is not satisfied [3]. In NDN, the router maintains an entry for every forwarded Interest in the Pending Interest Table (PIT) as long as the Interest's lifetime has not expired. During this time, the forwarding strategy can decide to retransmit if the Interest packet is not satisfied by a Data packet or when the strategy asks to explore and probe additional faces. While strategies such as *ncc*, the implementation of the default CCNx strategy in NDN, retransmit unsatisfied Interests, other strategies like *best-route*, the default strategy in NDN [4], leaving the application with the decisions of whether and when to retransmit.

When an application configures its namespaces to use a specific forwarding strategy, it couples its retransmission policy with the strategy mechanisms. Therefore a change in the strategy mechanism could impact the application. While a change can work well in isolated environments, where the change is application-specific, it can create conflicts when multiple applications use the same strategy. In addition, in shared resource environments such as core networks, it is unlikely that the application developer would have the freedom to select the strategies used.

While some applications require or could benefit from in-network retransmission, others request complete control of their traffic and avoid network retransmissions. Therefore we suggest that the decision to perform in-network retransmission should be made by the application, and executed by the forwarding strategy. We propose a simple forwarding strategy mechanism that allows applications to decide whether a network retransmission is required, and helps forwarding strategies to differentiate a network retransmission from an application retransmission.

2. RETRANSMISSION MECHANISM

The Interest and Data packets in NDN use type-length-value (TLV) encoding to represent each exchanged value. This provides an easy and dynamic platform for adding new information to either the Interest or the Data packet. Our suggested retransmission mechanism performs two independent yet complementary functions, application abstraction and retransmission differentiation.

First, we suggest adding a new *Interest Retransmission Policy (IRP)* TLV to the Interest packet to specify the application retransmission policy. The IRP TLV is a flag that indicates whether the application expects network retransmission, or whether it requires control within the application scope. By using IRP, the application can determine the policy for every Interest it sends. A forwarding strategy supports this policy by providing two retransmission mechanisms as part of its implementation, one that supports in-network retransmissions and another that supports application retransmissions. Algorithm 1 presents a simplified framework for a forwarding strategy that supports both retransmission mechanisms by checking the IRP flag when receiving an Interest.

The IRP flag does not determine the in-network retransmission algorithm: it requires only that one exists. Therefore, the application decides whether an Interest is retransmitted by the network, while the strategy determines the in-network retransmission algorithm, that is, when to retransmit and which next hop(s) to choose. In addition, different strategies can choose different retransmission algorithms according to network characteristics. For instance, it may be that a core network strategy would choose a retransmission algorithm that addresses congestion issues and relies on collecting round-trip-times, while an access strategy retrans-

ICN'16 September 26-28, 2016, Kyoto, Japan

ACM ISBN 978-1-4503-4467-8/16/09.
DOI: http://dx.doi.org/10.1145/2984356.2985241

```
Function ForwardInterst(interest):
    face_list = SelectNextHop(interest)
    IRP = GetIRP(interest)
    SendInterest(interest, face_list)
    if IRP then
        schedule retransmission at time x
    else
        wait for application retransmission
    end
    return
```

Algorithm 1: Application Abstraction Framework of a Forwarding Strategy

mission algorithm would simply follow a list of given faces and retransmit an Interest after a fixed time interval.

Second, we suggest adding a second Interest TLV, the *'Network Retransmission' (NR)*, to differentiate application Interests from network retransmissions. Using the NR TLV, strategies can support different mechanisms for control and data traffic, and collect performance measurements of alternative next hops in dynamic environments.

Figure 1 illustrates a problem that could be solved by using the NR TLV. In this scenario, R1 selected f1 as the best performing face before a new and faster path to R4 through R3 was added. To explore the RTT of the new path, an adaptive forwarding strategy such as *ncc* retransmits an Interest on f2 after forwarding it on the existing working f1. If the time between t1 and t2 is greater than 1/2(RTT1-RTT2), the Interest would arrive at R4 through f3 before f4. In this case R4 recognizes the second Interest from R3 as a duplicate Interest due to the same Nonce. Therefore R4 drops the second Interest and replies with NACK to R3. Here the strategy would fail to learn that f2 is a better performing face.

This problem can be solved by adding NR TLV to the Interest, and using it to avoid loop-detection NACKs and support face probing. In our early implementation, presented in Algorithm 2, we used a non-negative-integer to represent NR TLV and increased its value by one every time the Interest was retransmitted by the strategy. By using the Interest nonce, and by limiting the maximum value of NR, the strategy avoids forever forwarded Interests and differentiates application Interests from network retransmissions.

```
Function DetectLoopAndRetransmissions(interest):
    if nonce previously recorded then
        if NR == 'MaxAllowed' OR NR == 0 then
            send NACK
        else
            interest.NR++
            HandleRetransmission(interest)
        end
    else
        interest.NR++
        ForwardInterst(interest) [algorithm 1]
    end
    return
```

Algorithm 2: Retransmission Differentiation using NR

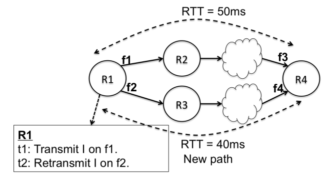

Figure 1: Loop-Detection NACK

3. PRELIMINARY RESULTS

We implemented the proposed retransmission mechanism in NFD 0.4 and tested it using the emulated NDN testbed in the open Network Lab (ONL) [5]. We modified the best-route strategy to use the proposed mechanism, and named the modified strategy as best-route-r. We ran one consumer and two producers on three of the testbed hosts, and configured the consumer to send 50 Interests per second and one producer to halt for 10 seconds out of 30 seconds total. We collected the numbers of sent and received packets and found that the average unsatisfied Interest rate of best-route is 42.55% and the one of best-route-r is only 0.621%.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we argue that coupling the application namespace with the retransmission mechanism of a forwarding strategy can be easily interrupted when the selection of the forwarding strategy is overwritten by the network operator, or when the implementation of the forwarding strategy changes. Our proposal suggests decomposing this mechanism by adding two new TLVs to every Interest packet: The IRP flag to support application abstraction, and the NR TLV to differentiate retransmitted Interests from others. Our preliminary results show that adding those two new TLVs can significantly improve the unsatisfied Interest rate of a multiple producers application running on an emulated NDN testbed. In future work we plan to further explore the details of the in-network retransmission mechanism, including the waiting time before retransmitting an unsatisfied Interest, and possible uses and representations of NR. In addition, we plan to evaluate the overhead created by supporting the suggested mechanism.

Acknowledgement

This work was supported by NSF grants CNS-1040643 and CNS-1345282 and by research gift from Cisco.

5. REFERENCES

[1] Lixia Zhang et al. Named data networking. *ACM SIGCOMM Computer Communication Review*, 2014.

[2] Project CCNx. http://www.ccnx.org/.

[3] Hila Ben Abraham et al. Forwarding strategies for applications in named data networking. In *Proceedings of the 2016 Symposium on Architectures for Networking and Communications Systems*, pages 111–112. ACM, 2016.

[4] Alexander Afanasyev et al. Nfd developer's guide. Technical report, NDN-0021, NDN, 2014.

[5] Ze'ev Lailari et al. Experiments with the emulated ndn testbed in onl. In *Proceedings of the 2nd International Conference on Information-Centric Networking*, pages 219–220. ACM, 2015.