Adaptive Duplicate Suppression for Multicasting in a Multi-Access NDN Network

Saurab Dulal
University of Memphis
sdulal@memphis.edu

Lan Wang
University of Memphis
lanwang@memphis.edu

ABSTRACT

This poster presents our ongoing work on adaptive duplicate suppression for multicasting in a multi-access NDN network. It includes our design, implementation, and some preliminary evaluation results. Our early evaluation shows a substantial reduction in duplicate traffic in NDN multicast communication.

CCS CONCEPTS

• **Networks** → **Network protocol design**.

KEYWORDS

Named Data Networking (NDN), Multi-Access Networks, Duplicate Suppression

ACM Reference Format:
Saurab Dulal and Lan Wang. 2022. Adaptive Duplicate Suppression for Multicasting in a Multi-Access NDN Network. In *9th ACM Conference on Information-Centric Networking (ICN '22), September 19–21, 2022, Osaka, Japan.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3517212.3559480

1 INTRODUCTION

Multicasting [8] simultaneously disseminates information to the members of a multicast group. It is an efficient choice for multi-party (one-to-many and many-to-many) communication in modern applications such as live video streaming, video conferencing, online gaming, vehicular networks, and disaster management. It can also eliminate the need for external infrastructures such as access points in wireless networks [9].

Named Data Networking (NDN) [10] provides native support for data multicast. In NDN, every piece of content is named, signed, and optionally secured at its creation. This enables the decoupling of the packets from their producers. Any intermediate node can cache and serve the data because the receiver can verify data provenance through the signature. These features help to achieve efficient and secure data distribution required by applications. However, the current NDN forwarder lacks a duplicate suppression mechanism for multicast in a multi-access network, which can cause network congestion and significantly degrade the overall packet delivery performance. Li et. al. proposed a leader based interest suppression scheme [4]. It chooses a client as the leader to forward an interest before the others. However, the leader can be a bottleneck and the solution is not adaptive to different network conditions.

In this poster, we present an *adaptive duplicate suppression scheme* for multicasting NDN Interests and Data packets in multi-access networks such as WiFi and Ethernet subnets. Each node constantly monitors the network and stores the count of duplicates per Interest and data flowing in the network. It uses the count to dynamically compute the suppression time, i.e., the maximum wait time per prefix before forwarding an Interest or data packet. Furthermore, we use a backoff algorithm [2] to compute the actual suppression timer (Section 3.3) based on the suppression time value. Preliminary evaluation shows that our design can significantly reduce redundant network traffic and adjust to different network conditions.

2 PROBLEM STATEMENT

Under multicasting, we can run into scenarios where lots of redundant traffic can flow across the network. Let us understand the problem with the help of Figure 1. At first, **node A** floods a subnet with an Interest (I = $/file/cat/jpg$) at time t=0ms. When the Interest is still pending, shortly after Δt another **node D** floods the subnet with the same Interest. This problem gets exacerbated when the number of consumers increases, and becomes worst if they all send the same Interest. A similar case can occur for the corresponding Data packet. For example, if every node in the network except the requester has the data, all of them can reply at once, as shown in Figure 1(b). Thus, duplicate traffic without suppression will sharply increase network congestion and bandwidth consumption [1]. It can significantly degrade the overall packet delivery performance and cause serious scalability issues.

One of the naive approaches to tackle this problem is random wait [9]. In this technique, a node waits for a random period of time before forwarding an Interest or data. If the same Interest or data is overheard during the wait, node drops the forwarding. However, this approach fails to adjust automatically along with various network conditions. For example, if a network is lossy, it is better to permit some duplicates, but for a stable network, allowing

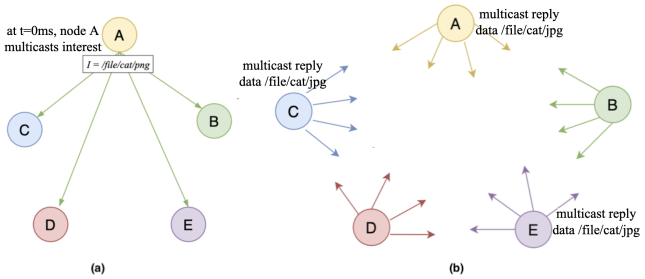

Figure 1: Flow of redundant traffic in a wireless network

ICN '22, September 19–21, 2022, Osaka, Japan

ACM ISBN 978-1-4503-9257-0/22/09…$15.00
https://doi.org/10.1145/3517212.3559480

zero duplicates is optimal. Thus, before forwarding an Interest or data packets, nodes should dynamically adjust their suppression times based on network conditions.

3 SYSTEM DESIGN

We categorize the suppression logic into two high-level cases: i) Look Behind and ii) Look Ahead.

3.1 Look Behind Case

In the look behind case, each node records the duplicate count of an Interest/Data packet sent to or received from a multicast face into a measurement table. For any Interest or Data that has an entry in the table, if the same is received again, the duplicate count for that entry is incremented. Every entry in the table has a short lifetime, on the order of the propagation delay in the multi-access network. Interest entries are removed from the table if they are satisfied or expire, whereas Data entries are only removed when they expire. Before a node sends an Interest or Data packet to a multicast face, it checks the measurement table for a corresponding entry. If this entry exists, the forwarding is dropped. Otherwise, the suppression logic goes to the Look Ahead case.

3.2 Look Ahead Case

In this case, nodes wait for some time (i.e., suppression time) before forwarding. While waiting, if the same Interest or Data packet is overheard, the forwarding is dropped. The suppression time should be adaptive to network conditions. For example, if the number of duplicates is high, the suppression time should be higher and vice-versa. Additionally, for lossy links, it should be reasonable to permit a few duplicates, and thus the suppression time can be smaller.

3.3 Adaptive Suppression Timer

The goal of the Adaptive Suppression Timer (AST) is to maintain the Interest and Data duplication counts below a certain threshold (e.g., 1 or 2), configured based on the loss rate of the environment.

Every time an entry is removed from the measurement table, we compute the exponential weighted moving average (EWMA) [3], similar to TCP, of the duplicate count (c) for the corresponding Interest or Data prefix. We use the following formula to compute the EWMA (e):

$$e_i = \begin{cases} c_1 & i = 1 \\ a * c_i + (1 - a) * e_{i-1} & i > 1 \end{cases} \quad (1)$$

Where $0 < a < 1$ = *smoothing factor*, i = *instance*, c = *duplicate count*. We use $a = 0.125$.

Next, we use EWMA to dynamically adjust the suppression time for a prefix, as shown in Figure 2. We create three different phases for the suppression time to achieve an optimal operating range:

Phase 1: Exponential Increase The suppression logic enters this phase if the EWMA is both above the duplicate threshold and increasing. This means that nodes did not wait long enough before forwarding, and are aggressively duplicating the same Interest or Data packets. Thus, we increase the suppression time (t) by some constant factor m, i.e. $t_i = t_{i-1} * m$.

Phase 2: Do Nothing If the EWMA is constant but above the duplicate threshold, the suppression logic enters the Do Nothing phase. As the name suggests, the suppression time is carried over from the previous instance. i.e. $t_i = t_{i-1}$.

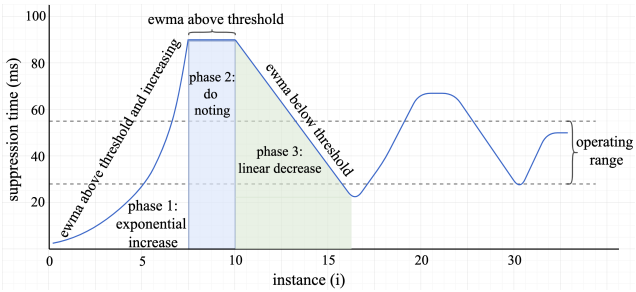

Figure 2: Phases of suppression time

Table 1: Average number of received packets per node with and without our suppression module (sup = suppression)

average number of received packets	w/o sup at producer	w/ sup at producer	w/o sup at consumer	w/ sup at consumer
Interest	2076	1020	1384	680
Data	0	0	1130	976
Unsolicited Data	0	0	473	318

Phase 3: Linear Decrease The suppression logic enters this phase if the EWMA is below the threshold and decreasing. This phase signifies that the assigned suppression timer is sufficient for nodes to not duplicate the same Interest or Data. However, it may not be optimal. Thus, we decrease the suppression time by some value n i.e. $t_i = t_{i-1} - n$.

The suppression logic constantly switches between these phases to dynamically adjust each suppression time until an optimal operating range for the given network condition is discovered. Finally, we compute the adaptive suppression timer (s) using the following formula:

$$s_i = rand(0, t_i) \quad (2)$$

4 PRELIMINARY EVALUATION

We implemented our algorithm in the NDN Forwarding Daemon (NFD) [6] and evaluated it using Mini-NDN Wifi [7]. We used a simple 4-node topology with 1 producer and 3 consumers connected via a wireless link. The producer published a 5MB (661 segments) file using putchunks [5], and all the consumers simultaneously fetched the file over the multi-access link using catchunks. We used a duplicate threshold of 1.5, and $m = 1.3$ and $n = 10$ for Phase 1 and 3, respectively. Table 1 shows that, with suppression, on average more than 50% and 13% of duplicate Interest and Data were suppressed. We also observe a 33% reduction in the number of unsolicited Data packets (a Data packet is unsolicited if the node receiving it did not send the corresponding Interest packet).

5 FUTURE WORK

As the next step, we will perform additional experiments using various network conditions and sizes, and compute other metrics such as delay, loss, and bandwidth consumption. We will also compare our algorithm with a few existing proposals. Additionally, we plan to use machine learning to tune the parameters for computing the suppression time.

6 ACKNOWLEDGMENT

This work was supported by the National Science Foundation award CNS-1629769.

REFERENCES

[1] Saurab Dulal. NDNSD: Service publishing and discovery in NDN. *University of Memphis Thesis*, 2020.
[2] Jonathan Goodman, Albert G Greenberg, Neal Madras, and Peter March. Stability of binary exponential backoff. *Journal of the ACM (JACM)*, 35(3):579–602, 1988.
[3] J Stuart Hunter. The exponentially weighted moving average. *Journal of quality technology*, 18(4):203–210, 1986.
[4] Menghan Li, Dan Pei, Xiaoping Zhang, Beichuan Zhang, and Ke Xu. Interest-suppression-based ndn live video broadcasting over wireless lan. *Frontiers of Computer Science*, 11(4):675–687, 2017.
[5] NDN Project Team. NDN Essential Tools. https://github.com/named-data/ndn-tools. (Accessed on Accessed on 08/13/2022).
[6] NDN Project Team. NFD: Named Data Networking Forwarding Daemon. https://github.com/named-data/nfd. (Accessed on Accessed on 08/13/2022).
[7] NDN Project Team. Mini-NDN – a lightweight NDN emulator. http://minindn.memphis.edu/, 2020. Accessed: 2022-08-13.
[8] Sanjoy Paul. *Multicasting on the Internet and its Applications*. Springer Science & Business Media, 1998.
[9] Podder, Proyash, and Gupta, Somak Datta and Neishaboori,Azin and Afanasyev, Alex. sV2Pc: On Scaling LTE-based Vehicle-to-Pedestrian Communication using NDN. https://www.nist.gov/news-events/events/2021/10/ndn-community-meeting-2021. (Accessed on Accessed on 08/29/2022).
[10] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, KC Claffy, Patrick Crowley, Christos Papadopoulos, Lan Wang, and Beichuan Zhang. Named data networking. *ACM SIGCOMM Computer Communication Review*, 44(3):66–73, 2014.