Multicast Distance Vector Forwarding in NDN Networks

Hin Ching Hou, Bradford Lowe
UCLA

Abstract
Multicast forwarding in Named Data Networking (NDN) with distance-vector routing presents a challenge, as current implementations often rely on inefficient broadcast approaches. This research investigates alternative multicast forwarding strategies to improve network efficiency in NDN-dv. A custom-built Python simulator was developed to evaluate different strategies, comparing a baseline flooding approach with a lowest-cost "greedy" approach. The simulation results compare the correctness and performance of both methods. While the simulator currently has some limitations, this work lays the foundation for future testing of the two strategies outlined or future strategies to be developed.

Contents

1 Introduction

Rapid computing and communication technology advances have fueled networked applications, especially those running on mobile devices communicating over wireless connectivities. These advancements have led to an explosion in data-intensive applications such as video streaming, online gaming, content distribution, and Internet of Things (IoT) deployments. Multicast, the efficient delivery of data to multiple receivers simultaneously, plays a crucial role in supporting these applications. Communicating entities at the network edge require efficient and scalable data delivery, often over dynamic and intermittent connectivities. In this context, effective multicast solutions are essential for optimizing network performance and resource utilization.

Named Data Networking (NDN) represents a promising direction to meet the above requirements by its direct use of application data names in communication, thus eliminating the need for translating DNS names to IP addresses; by its built-in security support that secures named data objects directly, thus removing the dependency on intermediate channels' security; and by its data-centric design that supports in-network

caching and delay/disruption-tolerant communications. All the above desired NDN functions are direct results of NDN's network model: a networked system is made of named entities with various trust relations among each other, where these named entities can be devices, servers/services, app instances, or anything that produces and/or consumes named packets. Since the names of these entities are decoupled from their specific attachment points to the network in general, they can explore any available connectivities to communicate.

The above picture differs fundamentally from IP's view that a network is made of interconnected nodes identified by IP addresses, with no security relations among IP nodes at the IP layer. However, NDN's view on networking also raises questions that must be answered before an entity can be effectively deployed in an NDN network.

NDN Sync protocols are required to synchronize the datasets of multiple users operating on the same data, such as a collaborative text editor. State Vector Sync (SVS) is the latest protocol widely used by NDN in this regard [3], and it requires multicast functionality within a sync group in order to be scalable.

In this work, we address the challenge of efficient data delivery in NDN, specifically focusing on multicast forwarding with distance-vector routing. Current multicast implementation in ndn-dv is essentially broadcast, which is functional but not scalable. Distance-vector routing is desirable - much less routing overhead. NDN's stateful forwarding plane allows for better forwarding strategies than IP distance-vector, more information in the network [5]. We explore improved multicast forwarding strategies within the NDN-dv context. We evaluate different forwarding strategies using a custom-built simulator, comparing a baseline flooding approach with alternative methods. This work provides a foundation for developing more efficient multicast solutions. First, we clarify the differences between the goal and process in multicasting a packet in an IP network and multicasting a packet in an NDN network. Second, we create a Python-based simulator for evaluating NDN-dv multicast forwarding strategies. Third, we evaluate two alternative forwarding strategies (lowest-cost greedy and shortest path) in comparison to the current strategy of flooding the network. We analyze of the challenges and trade-offs associated with distance-vector multicast forwarding in NDN. Fourth, we discuss future research directions.

2 Multicast Forwarding

2.1 Multicast Forwarding in TCP/IP Networks

Multicast forwarding in TCP/IP networks aims to deliver a single data packet to a group of interested receivers. This is more efficient than sending individual copies to each receiver (unicast) or sending the packet to everyone (broadcast). Several multicast routing protocols have been developed for IP networks, with DVMRP (Distance Vector Multicast Routing Protocol) being an early example that uses a distance-vector approach.

DVMRP builds source-based trees for multicast forwarding. In essence, for each source sending a multicast packet, a separate distribution tree is calculated. Routers using DVMRP maintain distance information to other routers, similar to distance-vector unicast routing. However, DVMRP uses this information to determine the reverse path back to the source.

When a multicast packet arrives at a DVMRP router, the router checks if it is on the shortest path back to the source. If it is, the packet is forwarded to all interfaces except the one it arrived on (Reverse Path Forwarding - RPF). This helps to prevent looping. If a router has no downstream neighbors that are part of the multicast group, it sends a prune message towards the source to trim branches of the dis-

tribution tree where there are no receivers.

2.2 NDN forwarding

NDN forwarding operates through the exchange of Interest and Data packets. A consumer sends an Interest packet for a specific named piece of data[1]. Routers forward this Interest packet towards potential data producers. Each router maintains a Pending Interest Table (PIT) that stores information about any Interests it has forwarded recently or is waiting to be forwarded[5]. When a router receives an Interest, it checks its Content Store (cache) for the requested data. If the data is found, it is returned in a Data packet. If not, the router forwards the Interest based on its forwarding strategy.

NDN naturally lends itself to multicast, unlike IP. If multiple consumers request the same data, the network efficiently serves these requests. Interest packets can be aggregated at each router. If multiple consumers request the same data, the router may receive multiple Interest packets with the same name. Instead of forwarding each Interest individually, the router can record the incoming interfaces in the PIT and forward only one Interest packet, dropping any copies of it. When the Data packet returns, the router uses the PIT information to forward the Data packet on the reverse path of the Interest. This aggregation mechanism helps to reduce network traffic and improve efficiency, effectively achieving multicast without the complexity of dedicated multicast protocols such as DVMRP.

2.3 NLSR

NLSR (Named-data Link State Routing) is the most widely-used routing protocol for NDN routers[2]. It adopts a link-state approach, where each router maintains a comprehensive map of the network topology. Routers exchange information about their directly connected neighbors and the links to them, enabling each router to independently calculate the shortest path to any destination. While NLSR offers efficient unicast and multicast routing capabilities, it represents a different paradigm compared to distance-vector routing.

2.4 NDN-dv

Distance-vector routing, as discussed earlier, is a routing protocol where each router maintains a table of the distances to other routers in the network through each neighbor. Routers exchange this distance information with their neighbors periodically and whenever the minimum distance to a node changes. This reduces the amount of network routing traffic compared to link-state, which sends much more frequent updates. In traditional TCP/IP networks, distance-vector routing faces challenges such as the count-to-infinity problem and slower convergence compared to link-state routing. These limitations have often made link-state protocols the preferred choice for IP networks. However, NDN's architecture provides a more favorable environment for distance-vector routing[4]. NDN's stateful forwarding plane, particularly the use of the Pending Interest Table (PIT), allows for enhancements to distance-vector forwarding that are not possible in IP. The PIT provides additional information about pending Interests, which can be leveraged to make more informed forwarding decisions and mitigate some of the drawbacks of distance-vector routing[5].

While ndn-dv does use SVS like NLSR, it is currently forced to run on a broadcast-like framework to send Sync Interests to a sync group, which is inefficient. Our goal is to improve this through better forwarding strategies.

3 Methodology

This section outlines the methodology used to evaluate different multicast forwarding strategies in NDN-dv. It describes the simulation

environment, the forwarding strategies implemented, and the evaluation metrics employed.

3.1 Simulation Environment

To evaluate the performance of multicast forwarding strategies, a custom-built simulator was developed using Python. Initial attempts to utilize existing NDN tools such as miniNDN and direct implementation in the ndnd library proved unwieldy prior to confirming the efficacy of a new forwarding strategy. MiniNDN lacks support for distance-vector routing, which is essential for this research. The goal of this research is to find a better forwarding strategy compared to ndnd's current implementation. Therefore, a new simulator was designed and implemented to provide the necessary functionality and control for conducting the experiments.

The simulator operates by simulating a chosen strategy over all nodes. It simplifies the current ndn-dv design for a simulation environment by using all the information in the RIB as forwarding information rather than a traditional FIB. The simulator allows for either hard-coded routing information for smaller test cases, or use of a Bellman-Ford routing simulator for larger cases. The simulator measures the number of interests produced (which we set up ourselves), interests sent, interests dropped, and interests received (not dropped) by each node. It also detects when an interest makes its way to all nodes, or when by the end of the routing some group member has not received it, to ensure correctness as well as performance.

3.2 Forwarding Strategies

To investigate the effectiveness of different multicast forwarding strategies, three distinct forwarding strategies were implemented within the simulation environment. The first strategy, flooding (broadcast), serves as the baseline for comparison. In this approach, each node, upon receiving an Interest packet, forwards it to all neighboring nodes it has not yet received that same packet from. While simple to implement, flooding often leads to significant redundancy and network congestion.

The second strategy, the lowest-cost greedy approach, aims to reduce the number of transmissions by using more of the information in the RIB. Specifically, a node employing this strategy forwards an Interest packet only to the neighboring node that has the lowest-cost path to a group member. This approach seeks to minimize the overall cost of reaching all receivers, with cost defined as the number of hops. However, it was quickly realized that this approach does not lead to correct outcomes, as packets do not reach all group members in many situations. Thus, this approach was dropped.

The third strategy, a modified lowest-cost algorithm, builds on the previous approach by incorporating additional logic to prevent redundant transmissions and to maintain correctness. Similar to the lowest-cost greedy strategy, this method forwards Interests along lowest-cost paths to group members. However, instead this strategy forwards to *all* neighbors which it has not yet received the interest from, and also lie on a lowest-cost path to *any* group member. This approach is the current focus of the study, and test results up to this point are promising.

3.3 Evaluation Metrics

The evaluation of each forwarding strategy was carried out by measuring several key characteristics of the network traffic. These metrics provide a quantitative basis for comparing the efficiency and overhead associated with each approach. Specifically, the following metrics were recorded for each node in the simulation: 1. the number of packets produced, representing the initial Interests generated by the node; 2. the number of packets dropped, indicating Interests that were discarded by the node due to the forwarding

strategy or other factors; 3. the number of packets kept, representing Interests that were processed by the node but not forwarded; 4. the number of packets sent, reflecting the total number of Interests transmitted by the node to its neighbors. Additionally, the main measure of correctness is tracked: whether or not all group members receive every produced interest.

To assess the overall effectiveness of each forwarding strategy, the total values for each of these metrics were aggregated across all nodes in the simulated network. By comparing these aggregate values, it is possible to determine which strategy minimizes network overhead (e.g., sent packets) while ensuring efficient delivery to all intended receivers.

4 Results

4.1 Results

The simulation was conducted across a variety of network topologies. For each topology, a set of group members and interest producers were randomly selected. To ensure a representative sample, we chose min(1000, number of subsets with at least 2 members) combinations of group members in each topology. This way, smaller test cases are exhaustively tested, and larger test cases were tested via a (non-exhaustive) random sample. Each of the two forwarding strategies was evaluated on each combination.

The primary results focus on the aggregate metrics for each strategy: total packets produced, total packets dropped, total packets kept, and total packets sent. Additionally, the correctness of each strategy was assessed by verifying whether all group members received every produced interest.

Across all tested topologies and group member combinations, the modified lowest-cost strategy outperformed the flooding strategy. Specifically, the modified lowest-cost strategy achieved the lowest number of total packets sent while ensuring that all group members received all produced interests. The lowest-cost greedy strategy, while reducing the number of sent packets compared to flooding, often failed to deliver interests to all group members.

The tested topologies include the Geant, Caida, Sprint, and NDN Testbed topologies, as well as some smaller topologies specifically created during the initial testing of the correctness of the simulator (topo1, topo2, topo3). So far, none of the runs we did on any topology have yielded an incorrect output. More testing on this end is required with larger, more varied topologies, so we will focus on the performance here.

In the following table, one run on each topology is detailed. We leave off the number of interests dropped and received for brevity, as total sent is a good enough measure. Each test only has 1 interest produced.

For the flooding strategy:

Topology	Nodes	Group Size	Interests Sent
Topo1	7	3	8
Caida	10	3	24
Testbed	36	10	94
Geant	44	7	53
Sprint	51	6	54

For the modified lowest-cost strategy with the same parameters:

Topology	Interests Sent	Improvement
Topo1	5	+3
Caida	14	+10
Testbed	81	+13
Geant	34	+19
Sprint	25	+29

Looking at these results, as the ratio of group size to total nodes decreases there is more of an improvement when using the lowest-cost strategy. This implies that the strategy should be substantially more scalable than flooding.

Next, we look at the average performance for one topology-group configuration between the two strategies. We choose the Sprint topology (51 nodes) with 7 static group members, then run using one interest starting at each group member, and add up the results. (Note: kept = received and not dropped).

Strategy	Dropped	Kept	Sent
Flooding	203	181	384
Lowest-Cost	123	112	235

The lowest-cost strategy is not able to do away with dropped interests, but it does minimize them compared to flooding.

Another interesting statistic is that 25 nodes never received an interest with the flooding strategy, whereas 34 nodes never received a single interest with modified lowest-cost. This highlights the lowest-cost strategy's ability to avoid sending to some extraneous nodes entirely compared to flooding, a key consideration in multicast forwarding.

4.2 Analysis

The results demonstrate a clear hierarchy in the performance of the three forwarding strategies.

Flooding, while guaranteeing delivery, incurs the highest overhead due to its indiscriminate forwarding of Interest packets. This leads to a large number of redundant transmissions and significant network congestion.

The **lowest-cost greedy approach** attempts to reduce overhead by forwarding only along lowest-cost paths. However, its limited forwarding scope often results in Interests not reaching all designated group members, indicating a correctness issue.

The **modified lowest-cost strategy** achieves the best balance between overhead reduction and delivery guarantee. By forwarding along lowest-cost paths while avoiding redundant transmissions, it minimizes the number of sent packets while ensuring that all group members receive the intended Interests. The additional logic to prevent redundant transmissions is crucial for its efficiency.

Note: More testing is required to prove that the modified lowest-cost strategy is correct in all scenarios. Tests up to this point lead us to be optimistic about the correctness, but we cannot yet be certain.

The data suggests that utilizing more information from the RIB, as done in the modified lowest-cost strategy, leads to more efficient multicast forwarding. By considering the lowest-cost paths and avoiding unnecessary retransmissions, this strategy improves the data dissemination process.

5 Discussion

5.1 Interpretation of Results

The findings of this study highlight the importance of informed forwarding decisions in multicast routing. Simple flooding, while easy to implement, is inefficient and does not scale well. Strategies that consider network topology and path costs, such as the modified lowest-cost strategy, can significantly reduce network overhead.

The failure of the lowest-cost greedy approach underscores the need for a balance between minimizing transmissions and ensuring complete delivery. A strategy that is too restrictive in its forwarding may fail to reach all intended receivers.

The success of the modified lowest-cost strategy demonstrates the potential of leveraging information available in NDN's stateful forwarding plane to improve multicast efficiency in a distance-vector environment. The PIT, which stores information about pending Interests, enables strategies to avoid redundant transmissions and enhance overall performance.

5.2 Comparison of Strategies

As expected, the modified lowest-cost strategy outperformed the other two strategies. The flooding strategy served as a baseline, showing the highest overhead. The lowest-cost greedy strategy demonstrated that while using RIB information can reduce overhead, it's not enough to guarantee correctness. The modified lowest-cost strategy built upon the lowest-cost greedy approach to achieve both reduced overhead and correctness.

5.3 Limitations of the Study

This study has several limitations that should be considered when interpreting the results.

Topology Selection: The network topologies used in the simulations were limited in number and complexity. While they provided a basis for comparing the forwarding strategies, they likely do not fully represent the diverse range of network topologies found in real-world deployments. A more comprehensive evaluation would involve testing on a larger and more varied set of topologies, including much larger randomly generated ones.

Static Network: The simulations assumed a static network environment with fixed links and no link failures. In reality, networks are dynamic, and link failures can occur. The forwarding strategies' robustness to such dynamic changes was not evaluated in this study.

Routing Algorithm: The simulator uses a simplification of the actual Bellman-Ford algorithm, although the actual RIB should not be any different. The interaction between the forwarding strategies and a dynamic routing protocol was not explored.

Traffic Patterns: The simulations used a simplified model of traffic patterns, by simply iterating over each node using the simulator and forcing all interests to be forwarded immediately, if available. More complex and realistic traffic scenarios could be used in future work.

Number of Group Members: While the number of group members was varied, the range of variation and the impact of group size on the performance of each strategy could be explored in greater depth.

5.4 Future Work

While the performance increase of the lowest-cost strategy is substantial compared to the flooding strategy, more work needs to be done. Other potential strategies should be looked at, and more testing of the current strategy is required. While the current results look optimistic, a formal proof or proof by exhaustive testing is needed to ensure the correctness of the lowest-cost strategy in all topologies. The simulator can be updated as previously outlined, and both sparse and dense connection environments must be tested, as well as larger topologies since the current testbed is much smaller than networks would likely be in a fully-realized NDN world.

6 Conclusion

In this work, we investigated the problem of efficient multicast forwarding in NDN-dv. Recognizing the limitations of the current broadcast-based approach, we explored alternative forwarding strategies with the goal of reducing network overhead while ensuring reliable data delivery.

To facilitate this investigation, we developed a custom simulation environment that allowed us to evaluate different forwarding strategies in a controlled setting. Using this simulator, we compared the performance of three strategies: flooding, lowest-cost greedy, and a modified lowest-cost approach.

Our results demonstrate that the modified lowest-cost strategy offers a significant improvement over both flooding and the basic lowest-cost greedy approach. By leveraging information from the RIB and PIT, the modified strategy effectively minimizes redun-

dant transmissions while, in the testing so far, guaranteeing delivery to all group members.

While this study provides valuable insights into multicast forwarding in NDN-dv, it also highlights several avenues for future research. These include enhancing the simulator with dynamic routing capabilities and support for link failures, exploring a wider range of network topologies, and investigating more advanced forwarding strategies that utilize NDN's stateful forwarding plane to its fullest.

Overall, this work contributes to a deeper understanding of the challenges and opportunities in designing efficient multicast solutions for NDN-dv. The findings suggest that distance-vector routing, when combined with NDN's stateful forwarding capabilities, can be a viable approach to achieving scalable and efficient multicast communication.

6.1 Link to code:

https://github.com/brad-lowe/ndn-dv-multicast

References

[1] Alex Afanasyev, Jeff Burke, Tamer Refaei, Lan Wang, Beichuan Zhang, and Lixia Zhang. A brief introduction to named data networking. In *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, pages 1–6, 2018.

[2] A K M Mahmudul Hoque, Syed Obaid Amin, Adam Alyyan, Beichuan Zhang, Lixia Zhang, and Lan Wang. Nlsr: named-data link state routing protocol. In *Proceedings of the 3rd ACM SIGCOMM Workshop on Information-Centric Networking*, ICN '13, page 15–20, New York, NY, USA, 2013. Association for Computing Machinery.

[3] Philipp Moll, Varun Patil, Lan Wang, and Lixia Zhang. Sok: The evolution of distributed dataset synchronization solutions in ndn. In *Proceedings of the 9th ACM Conference on Information-Centric Networking*, ICN '22, page 33–44, New York, NY, USA, 2022. Association for Computing Machinery.

[4] Varun Patil, Sirapop Theeranantachai, Beichuan Zhang, and Lixia Zhang. Poster: Distance vector routing for named data networking. In *Proceedings of the 20th International Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '24, page 23–24, New York, NY, USA, 2024. Association for Computing Machinery.

[5] Cheng Yi, Alexander Afanasyev, Ilya Moiseenko, Lan Wang, Beichuan Zhang, and Lixia Zhang. A case for stateful forwarding plane. *Comput. Commun.*, 36(7):779–791, April 2013.