Interest Packet Multicast for Named-Data Link State Routing Protocol

Lukas Trommer
lukas.trommer@ucla.edu
University of California Los Angeles
Los Angeles, California, USA

Yash Lala
yash.lala@ucla.edu
University of California Los Angeles
Los Angeles, California, USA

ABSTRACT

We propose an extension to the Named-Data Link State Routing protocol (NLSR) which allows for the multicast routing of Interest packets in a network using Named-Data Networking (NDN). The receivers of those multicast Interest packets form a group which is identified by a unique NDN name. Multicast group membership is advertised to other routers in the network via prefix Link State Announcements (LSA), whose packet format has been modified with this proposal to incorporate both unicast and multicast names.

CCS CONCEPTS

• **Networks → Routing protocols**.

KEYWORDS

networks, named-data networking, routing protocols, multicast

1 BACKGROUND

1.1 Named-Data Networking

Named Data Networking (NDN) is an alternate Internet layer 3 concept intended to address architectural weaknesses in the current Internet architecture. Rather than identifying packets by host IP addresses, NDN describes data directly by a name and allows applications to fetch it in a pull-based manner. The current NDN implementations incorporate a variety of different protocol and integration proposals, including the *NDN Forwarding Daemon* (NFD) [4] and the *Named-Data Link State Routing Protocol* (NLSR). NLSR is an application level, intra-domain routing protocol for NDN similar to many IP routing protocols, but using Interest and Data packets to disseminate routing updates and therefore directly benefiting from the built-in data authenticity of NDN [6]. NLSR routers propagate routing information via Link State Announcements (LSA) throughout the network. Each router maintains adjacency relations with its neighbors, encodes and disseminates this information in an *adjacency LSA* for deriving network topology information and advertises its own name prefixes via a *prefix LSA*. A more detailed explanation about those components can be found in the referenced resources.

1.2 Multicast

Multicast describes a communication scheme in which multiple destinations for information transmission are addressed simultaneously. In NDN, the concept of Interest and Data packets establishes a bidirectional communication pattern. If multiple nodes in the network express Interest in the same named data, those packets are routed to the respective data producer and may be aggregated within the network if an Interest packet reaches a network node where the Interest already exists in the *Pending Interest Table* (PIT). In this case, the PIT entry is extended with the incoming face that received the duplicate Interest and the packet is not forwarded any further. Whenever a Data packet reaches a network node that has a pending Interest with multiple faces for its name, the Data packet is forwarded via all the indicated links. Enabled by this architectural design of the stateful forwarding plane in NDN, Data packet multicast is naturally supported for all network nodes that express Interest in a certain unit of named data. The bidirectional communication pattern of NDN also induces the need for Interest packet multicast for certain applications such as in [2] and [3].

Data packets are forwarded throughout the network on the paths of Interest packets that solicited them. Interest packets, in contrast, reach the data producing nodes based on the forwarding decisions of each network node. Interest packet forwarding is typically influenced by routing decisions which populate the FIB and are generated by a routing protocol such as NLSR. The routing protocol evaluates the advertised names of each network node and uses topology information to calculate adequate transmission paths throughout the network.

We identify the following requirements for Interest multicast support throughout the network:

(1) Each network node with the aim of receiving Interest packets under a certain multicast name must advertise this name prefix throughout the entire network.
(2) Routing decisions must induce a forwarding path from each sending network node to each receiving node. No packet shall be forwarded via the same link more than once.
(3) If a routing decision indicates a path using multiple adjacent links of a network node, an Interest packet must be forwarded via multiple outgoing faces simultaneously.

Requirement 3 is covered by the forwarding strategy in NFD, which offers a multicast option to forward Interest packets to multiple next hops at the same time. The strategy can be set and unset name-wise using the respective commands of the NFD Management protocol [5]. Therefore, once a routing decision for a specific multicast name has been derived, it does not only need to be added to the FIB but also associated to the multicast strategy.

In the following sections, we refer to a multicast group member as a network node that advertises a certain multicast name prefix and a multicast group as a set of group members that advertise the same name prefix.

2 CONCEPT & IMPLEMENTATION

2.1 Name Prefix Type Distinction

A routing decision for a unicast name requires a lookup of one network node that announces this prefix followed by a route calculation

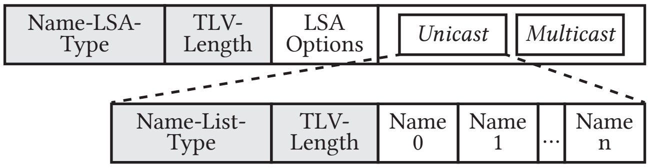

Figure 1: The packet format of a multicast-aware *prefix LSA*. The list of name prefixes is replaced by two name list blocks, one for unicast and one for multicast names.

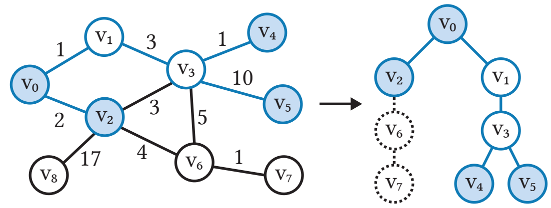

Figure 2: A network graph with multicast member nodes v_0, v_2, v_4 and v_5 (left) and the resulting spanning tree rooted in v_0 (right). The dotted tree branch has been pruned off.

to this node only. A routing decision for a multicast name requires a lookup of many network nodes that announce that prefix followed by the calculation of a shared route including all member nodes and the processing network node itself. Observing that routing along a group-specific shared graph including multiple target nodes is fundamentally different than routing to single nodes directly on the shortest possible path, a multicast extension of NLSR needs unicast and multicast names that are distinguishable from each other and processed separately. We identified the prefix-handling modules in the NLSR codebase and extended them with appropriate multicast-aware data structures. We furthermore updated the configuration file format with a new multicast entry type, the NLSR Management protocol to incorporate static and dynamic manipulation of advertised names with respect to both name types when configuring an instance of NLSR and the command-line configuration tool *nlsrc* to address those changes with new commands. As stated in point 1 of the requirements in section 1.2, the necessity of advertising those dedicated multicast names to the network emerges. We propose an updated prefix LSA format as shown in figure 1 which replaces the list of unicast names with two *name list* blocks, one for each prefix type.

2.2 Overlay Spanning Tree

We describe a network as an undirected graph with the routers represented as set of nodes, the links between routers as edges and the costs of each link as edge weights. Satisfying requirement 2, one has to find a subgraph for *each* multicast group, spanning over all respective group members, on which Interest packets are forwarded. In our implementation, we chose a shortest-path tree with a well-defined root node which is calculated using *Dijkstra's algorithm* [1]. As the algorithm iteratively adds nodes to the tree that is being built, we stop its execution once all member nodes of the group have been incorporated into the tree. By the nature of this method, building a shortest-path tree over the entire graph, the resulting tree might contain branches with leaves that are not multicast member nodes even if execution is stopped early. The dotted branch in figure 2, resulting from v_6 and v_7 being added to the tree before v_5, is an example for such behavior. Routing packets to those nodes introduces an unwanted traffic overhead that can be avoided pruning off the affected branches until only multicast members are leaf nodes. Figure 2 shows a sample network and the resulting tree as obtained by the described procedure.

The presented problem is a variant of the Steiner tree problem for graphs and other, potentially more efficient approaches have been outlined for example in [7].

The spanning tree computations are performed on each router in the network. Every router includes itself in the set of member routers to assure that it is part of the resulting spanning tree and appropriate routing decisions for packets can be derived in every situation, even if a router apart from the actual spanning tree received an Interest packet from a non-member node. As Interest packets are not identified by their sending node, an common tree root must be selected for all group members to assure obtaining the same spanning tree and in consequence equal routing decisions in all network nodes. The choice of the tree root node strongly influences the structure of the tree and therefore the effectiveness of the resulting forwarding routes. The problem of selecting a (nearly) optimal root node as input for the multicast tree construction algorithm is, however, not covered by the scope of this paper. For demonstration purposes, we use the static approach to choose the router with the alphabetically lowest name in the set of multicast member nodes.

3 EVALUATION

Our concept provides a simple implementation for Interest multicast based on an existing link-state routing algorithm. Naturally, a link-state routing protocol might induce scaling problems with a growing amount of nodes in the network or advertised multicast names. As this thought does not specifically address our implementation but the underlying routing protocol in general, we will not discuss potential drawbacks resulting from this architecture at this point.

We decided to change the format of the NLSR prefix LSA to incorporate unicast and multicast names in a single packet layout. While this decision enables Interest multicast with minimal changes to the protocol and the existing implementation of NLSR, it might be beneficial to separate both name types into separate LSA formats. If the expected update rates of the unicast and multicast name sets within a network differ substantially, a separation allows to install independent processing pipelines for both transmission types with different advertisement rates. Additionally, it might be beneficial to establish an alternative recalculation strategy for multicast trees which reuses previous computation results to account for group membership updates. Further research with respect to those aspects could therefore improve our conceptual implementation towards a reliable and efficient link-state multicast algorithm.

REFERENCES

[1] E. W. Dijkstra. 1959. A note on two problems in connexion with graphs. *Numer. Math.* 1, 1 (01 Dec 1959), 269–271. https://doi.org/10.1007/BF01386390

[2] Ashlesh Gawande, Jeremy Clark, Damian Coomes, and Lan Wang. 2019. Decentralized and Secure Multimedia Sharing Application over Named Data Networking. In *Proceedings of the 6th ACM Conference on Information-Centric Networking* (Macao, China) *(ICN '19)*. Association for Computing Machinery, New York, NY, USA, 19–29. https://doi.org/10.1145/3357150.3357402

[3] Tianxiang Li, Wentao Shang, Alex Afanasyev, Lan Wang, and Lixia Zhang. 2018. A Brief Introduction to NDN Dataset Synchronization (NDN Sync). In *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*. 612–618. https://doi.org/10.1109/MILCOM.2018.8599772

[4] NDN Project. 2022. NDN Forwarding Daemon (NFD). https://named-data.net/doc/NFD/current/. Online; accessed 2022-07-17.

[5] NDN Project. 2022. NFD Management Protocol. https://redmine.named-data.net/projects/nfd/wiki/Management. Online; accessed 2022-06-09.

[6] Lan Wang, Vince Lehman, A. Hoque, Beichuan Zhang, Yingdi Yu, and Lixia Zhang. 2018. A Secure Link State Routing Protocol for NDN. *IEEE Access* PP (01 2018), 1–1. https://doi.org/10.1109/ACCESS.2017.2789330

[7] Y. F. Wu, P. Widmayer, and C. K. Wong. 1986. A faster approximation algorithm for the Steiner problem in graphs. *Acta Informatica* 23 (05 1986), 223–229. https://doi.org/10.1007/BF00289500