The Need for a Name to MAC Address Mapping in NDN: Towards Quantifying the Resource Gain

Peter Kietzmann  
HAW Hamburg  
peter.kietzmann@haw-hamburg.de

Cenk Gündoğan  
HAW Hamburg  
cenk.guendogan@haw-hamburg.de

Oliver Hahm  
INRIA  
oleg@riot-os.org

Thomas C. Schmidt  
HAW Hamburg  
t.schmidt@haw-hamburg.de

Matthias Wählisch  
Freie Universität Berlin  
m.waehlisch@fu-berlin.de

ABSTRACT

In this paper, we investigate how broadcast in current solutions and design options impacts NDN scenarios. In experiments, we conduct micro benchmarks on low-end IoT devices to better understand the potentials of a dynamic MAC address-to-face mapping. Our results show that distributing L2 packets via unicast instead of broadcast leads to significant performance gains with respect to local system resources. The objective of this position paper is to be a discussion starter to find a common solution for efficient data distribution on both the network layer and the data link layer.

The remainder of this paper is structured as follows. In Section 2, we discuss the problem space and current solutions. For the case of dynamic face mapping, we introduce the design space in more detail in Section 3. We present our measurements in Section 4, and conclude in Section 5.
2 PROBLEM SPACE AND RELATED WORK

2.1 The IoT Use Case

The Internet of Things gathers a diverse set of very heterogeneous nodes. Our focus is on low-end IoT devices, equipped with hardware resources of class 2 [8], connected via radio, and powered by battery. These devices benefit from NDN in the following way. First, the lightweight NDN network stack requires less memory compared to the current IoT stack standardized in the IETF. It is worth noting that cache sizes are independent from network stack sizes. NDN provides off the shelf name-based management and monitoring capabilities, without introducing dedicated services on top of the network layer. Second, those devices may offload data to more powerful nodes without the burden of additional protocols. This is implemented natively on the network layer, and thus simplifies application programming. Third, this in-network caching allows nodes to sleep longer, reduce data delivery latency, and increase content availability [13].

It is worth stressing that the IoT does not only gather different nodes among different operators but may also be heterogeneous within a single domain. As such, we cannot assume that all nodes provide the same set of services, neither on the application, nor on the network layer.

2.2 Current Solutions and Challenges

2.2.1 NDN-specific Link Layer. Shi et al. [18], Grassi et al. [9, 10], and Wang et al. [20] argue for a link protocol that is specific to NDN. Shi et al. [18] introduce NDNLP, which features fragmentation and reassembly as well as acknowledgement and retransmission of packets. NDNLP is located between the network layer (e.g., NDN) and the virtual (e.g., tunnels) or physical (e.g., Ethernet) link layer. Grassi et al. [9, 10] present a link adaptation layer, which is tailored for vehicular networks but follows conceptually the same idea as NDNLP. Similarly in the context of improved reliability, Wang et al. [20] introduce an NDN broadcast protocol, which tries to minimize collision. Both approaches aim for an increased packet delivery ratio by measures below the application and network layer but still require packet processing by the CPU, independent whether a specific NDN service bound to the broadcast packet is available or not.

2.2.2 Name-based Filtering on NIC. Shi et al. [17] propose name-based filters on the device driver level of the network interface card. To optimize the implementation for limited on-chip memory of the NIC, names are maintained in a Bloom filter table. This approach exhibits good performance results but comes with the drawback of a layer violation. The data structure to implement filtering is specific to the ICN approach above the link layer. However, not all ICN approaches follow the same naming scheme [4, 21]. Consequently, changing the specific ICN network stack may require update of the device driver. This will slow down deployment of upcoming approaches. More importantly, this approach distributes data via layer 2 broadcast frames, and thus does not benefit from error handling on this layer.

2.2.3 Unicast Faces. Approaches different from the adaptation of the link layer or device driver are presented by Teubler et al. [19] and Baccelli et al. [7]. They introduce unicast faces. Basically, unicast faces assign (unicast) MAC addresses to NDN faces. These are created dynamically. Initial Interests are broadcasted, containing the unicast source MAC address of the sender. Having this information in place, the receiver makes use of the source address to assign a unicast face. NDN packets which are transmitted via a unicast face conversely include the unicast MAC address. This allows both native MAC-based filtering and benefitting from error handling/prevention on the data link layer. This approach is suitable for specific adaptations to link layers like TSCH [12], and in case not all nodes within a broadcast domain provide the same network layer services, such as in the IoT. On the other hand, unicast traffic reduces caching capabilities and data redundancy. A detailed analysis of link layer unicast and broadcast on the system load of an NDN node is still not present. In this paper, we argue that NDN should revisit the MAC-layer mapping. There are application scenarios in which a reduced system load outperforms data redundancy. Our analysis in Section 4 is a first step in this direction.

3 DESIGN SPACE BY INSTRUMENTING EXISTING LINK LAYER FEATURES

An NDN node can send Interest as well as data packets via unicast or broadcast on the MAC layer. In this section, we discuss pros and cons of each configuration scheme and perform a first experimental reality check about the effect of link layer support.

For the sake of clarity, we focus on core aspects. We assume NDN nodes with a single interface connected to the network via shared media. Extending this scenario to nodes with multiple interfaces does not change the core insights. Furthermore, we do not explicitly discuss multicast for two reasons. First, typical lower layer IoT protocols (e.g., 802.15.4) do not support multicast. Second, linking a face to a multicast MAC address instead of a unicast MAC address requires only a name to group address mapping on the data link layer.

3.1 Broadcast or Unicast for Interest or Data?

Case 1: Interest Broadcast, Data Broadcast. Any node within the broadcast domain will send Interests as well as data packets. As long as there is a matching name prefix in the forwarding information base (FIB), these (successfully received) Interests will create a PIT entry. Consequently, as soon as a corresponding data packet is transmitted within the broadcast domain, all members of this domain will forward this data packet, leading to redundant traffic. This highest level of redundancy has pros and cons. Practically, in a densely connected network any node may fail without degrading data delivery, as all remaining nodes cache content. However, this level of packet redundancy introduces excessive overhead for each node (e.g., CPU processing) but also for the complete network (e.g., radio interferences). Interferences should be considered even more seriously with
broadcast traffic, as there is no protective repair of errors on the link layer.

Case 2: Interest Broadcast, Data Unicast. Similar to case 1, all nodes of the broadcast domain will create a PIT entry after receiving an Interest packet. However, data packets are directed to the unicast MAC address which is associated with the corresponding outgoing face (i.e., the MAC address of the next hop). As we assume a shared media, all other nodes within radio reach receive the data packet, as well, but drop it at the device driver level because of an unmatching (unicast) MAC address. Those nodes can neither cache nor forward the data on the network layer. Previously created PIT entries will thus not dissolve by receiving data but by timeouts. These PIT entries require memory, processing time and will not help to achieve redundancy. To cope with node failures, an additional mechanism is needed to keep the MAC-face assignment in sync with the MAC address of an alternative next hop.

Case 3: Interest Unicast, Data Unicast. Compared to the previous scenarios, in this case, Interest as well as data packets are sent to a unicast MAC address, using unicast faces as described in Section 2.2.3. Such an approach implements hop-by-hop forwarding on the link layer and prevents redundancy completely because any overheard packet is dropped by the network interface card. This setup requires active maintenance of MAC-to-face mapping in case of node failures.

In contrast to Case 2, updating only the unicast data face is not sufficient. Data will be forwarded based on PIT entries. The strong coupling of Interest and data flow requires that the MAC address assigned to the Interest face is in line with the data face. However, usually there is a time gap between sending Interest and forwarding corresponding data. A unicast MAC address that is valid during Interest submission might be outdated when data is forwarded. On the other hand, this case reduces radio transmissions and CPU processing to a minimum and fully incorporates MAC layer retransmission handling.

Case 4: Interest Unicast, Data Broadcast. The last case provides very limited redundancy. Data packets will be processed by the NDN stack of all nodes of the broadcast range. However, as Interest has been delivered via MAC unicast, only one node in the broadcast domain created a PIT entry. All other nodes will thus drop the data packet at network layer.

Discussion. Case 1 promises path and data redundancy but comes to the cost of excessive resources consumption which may be harmful, especially in IoT networks. Case 2 optimizes data transport via unicast but keeps forwarding redundancy and superfluity of a routing protocol. Case 3 fully optimizes resource overhead and transmission robustness which is promising for battery driven, constrained nodes. However, this approach requires a reliable routing mechanism since it minimizes path redundancy as well caching capabilities. Case 4 brings little benefit to NDN, as redundant data is not utilized.

### 3.2 The Case for Link Layer Assistance

**Experimental Exploration.** In this initial experiment, we want to check back on the effect of a reliable unicast link layer by counting incomplete Interest-data handshakes. For this, we select three nodes within radio range from the Lille site of the FIT IoT-Lab testbed (s. Section 4). One consumer node requests 1000 content items from one producer node at a rate of two Interests per second (without Interest retransmissions), installing different MAC layer mappings. The third node generates side traffic on the same radio channel, sending packets of 50 Bytes within random intervals between 3-10 ms.

**Results.** Table 1 presents unsatisfied Interests at the consumer as an indicator of packet loss (Interest or data). Strikingly, we see that broadcasting Interests increases the error rate by about one order of magnitude (Case 2 versus Case 3). Broadcasting data after Interest unicasts appear more robust, which we account to an implicit link-layer coordination. In the presence of a periodic radio interferer, Interests are retransmitted on the MAC layer until the interferer paused and the transfer succeeded. Data in this single hop scenario follows immediately and thus takes advantage of the same pause (Case 2 versus Case 4). Broadcasting Interest and data (Case 1) combines these two sources of errors from Case 2 and Case 4. We conclude that NDN can substantially benefit from utilizing the support of MAC layer robustness.

<table>
<thead>
<tr>
<th>Case</th>
<th>Interest Unicast</th>
<th>Data Broadcast</th>
<th>Unicast</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>1.9 % (Case 1)</td>
<td>12.1 % (Case 1)</td>
<td>10.6 % (Case 2)</td>
</tr>
<tr>
<td>Case 2</td>
<td>3.3 % (Case 4)</td>
<td>19.9 % (Case 3)</td>
<td></td>
</tr>
<tr>
<td>Case 3</td>
<td>10.6 % (Case 2)</td>
<td>12.1 % (Case 1)</td>
<td>19.9 % (Case 3)</td>
</tr>
</tbody>
</table>

**Table 1: Unsatisfied Interests with different face to MAC address mappings under presence of link layer interference.**

### 4 EXPERIMENTAL EVALUATION

The objective of our experiments is the measurement of basic effects through different MAC layer mappings in a common IoT environment. Thereby, we make use of standard software solutions and typical IoT hardware including low power radio transmission technologies. We will focus on Case 2 and Case 3 (see Section 3) because CCN-lite does not support data transmissions via broadcast in the current version.

#### 4.1 Basic Testbed Setup

All experiments are conducted in the FIT IoT-LAB testbed [2] to reflect common IoT properties. The testbed consists of several hundreds of class 2 devices equipped with an ARM Cortex-M3 MCU, 64 kB of RAM and 512 kB of ROM, and an IEEE 802.15.4 radio (i.e., Atmel AT86RF231 [5]). The radio card provides basic MAC layer functions implemented in hardware, such as ACK handling, retransmissions, and CSMA/CA. For power measurements, we parameterize the consumption monitoring tool of the testbed with a conversion time of 332 μs and averaging over 64 samples. The
we analyze three basic configurations. (i) On all nodes, we install a common prefix route that covers all content names, and the corresponding face refers to the broadcast address. Note, in this case, a unicast MAC address conflicts with reachability of arbitrary content items via a single hop as content is requested from multiple producers. In the remaining configurations, we install dedicated FIB entries only on the selected consumer, which refer (ii) either to unicast addresses of the producers or (iii) to the broadcast address.

Furthermore, to analyze different network sizes and load, we vary the number of producer nodes, or the number of content items per producer in a predefined network size.

**Variable network size.** The number of content items per node is fixed but we increase the number of producers in different parameter settings. We implement a fixed average content request rate per producer, i.e., the number of Interests sent by the consumer increases linearly with the number of nodes in the network.

**Variable number of contents items per node.** The number of nodes is set to 20 and the number of content items per node is increased over different measurement runs. We apply a constant content request rate at the consumer.

### 4.2.2 Results.

Figure 2 shows the number of system wakeups per producer for the single hop scenario, with variable network sizes and a fixed number of content items per node (see Figures 2(a) and 2(b)), as well as with a fixed network size and variable number of content items per node (see Figures 2(c) and 2(d)). Figures 2(a) and 2(c) represent the setup where Interests are sent to the broadcast MAC address and all nodes have routing entries for all content names. Figures 2(b) and 2(d) represent the setup where only the consumer node has FIB entries, which maps faces either to the unicast address of each content producer or to the broadcast domain. Correspondingly, Figure 3 represents statistics of the CPU usage we measured.

In terms of energy and processing overhead, it is clearly visible that faces with unicast MAC addresses outperform broadcast faces. While the number of system wakeups is constant for varying network sizes, it only increases in direct relation to the number of provided content items of one node with unicast mapping. Broadcast overhead increases linearly with the number of nodes and the number of contents per node, thus it directly correlates with the total number of requested content items in the whole (single-hop) network. This increases resource consumption.

To summarize, unicast faces can improve the lifetime of battery driven IoT devices by keeping CPU-wakeups and processing overhead at a minimum, and increase stability by benefiting from build-in MAC layer mechanisms for unicast traffic, such as ACK handling and retransmission. On the other hand, it requires a maintenance mechanism for the assignment of MAC addresses to faces and reduces redundancy by omitting built-in content replica-mechanisms as well as alternative data paths.

Deploying common prefix routes to broadcast faces on all nodes reduces the overall performance of the network even more, as each node in the broadcast domain does not only wake up during incoming Interests, but also forwards Interest...
packets that will not be satisfied, as well as data packets that might be received as a consequence of the forwarding mechanism. This leads to an excessive number of wakeups of all nodes in the domain as well as additional data transmissions. The overhead in this broadcast scenario is several orders of magnitudes higher than that of unicast.

To better understand the overhead introduced by broadcast Interest forwarding when all nodes store FIB entries, Figures 2(a) and 2(b) present wakeups, which are separately shown for the radio device and the CCN-lite (CCNL) network stack. In a setup consisting of a single application, single network stack, and a single network interface, both measurements should be roughly equal as the link layer forwards each broadcast packet up to the network stack and vice versa. This holds only in case of a single forwarder (see Figure 2(b)). We detect a much higher number of wakeups by the device driver when all nodes store routes. The impact on CPU times is worse by a factor of 20, as depicted in Figure 3(a). We assume radio channel saturation causing this increased resource consumption. To further back these observation, we also measured (on the same network scale) (i) the rate of unsatisfied Interests (0 – 50 %), (ii) radio statistics from which we compute the rate of unsuccessfully transmitted packets due to failing CSMA/CA channel access (0.39 – 0.56), and (iii) the average number of network layer retransmissions (2 – 9 %). All these observables indicate a negative impact on network utilization while broadcasting.

Figure 2: Number of system wakeups for varying network setups.

Figure 3: Absolute CPU usage for varying network setups.

Figure 4 displays the energy per node additionally consumed when Interests are broadcasted. We show the energy excess of Interest broadcast, data unicast with and without Interest forwarding over an Interest unicast, data unicast mapping for varying network sizes. By no surprise, the graphs resemble Figures 2(a) and 2(b), and the additional consumption with forwarding exceeds single-hop broadcast by orders of magnitudes, for increasing (single-hop) networks.

4.3 Multi-hop Scenario

4.3.1 Configuration. We conduct our multi-hop measurements in Grenoble. This site of the testbed provides placement
of nodes such that nodes do not form a single broadcast domain. However, fluctuating properties of the wireless media (e.g., reflections) may lead to changing topologies from multi-hop to single-hop. To ensure a minimal multi-hop connectivity, we introduce a monitoring phase before our experiments start which is based on the mechanism proposed in [11]. During this phase, we identify a set of nodes that inter-connect over multiple hops. The resulting topology consists of 30 nodes where one node acts as consumer, 20 nodes act as producers with a distance of three hops towards the producer, and other nodes serve as intermediate nodes on the path.

Similar to the single-hop scenario, leaf nodes of the resulting topology are equipped with unique content items that are requested by a single consumer in randomized order. Referring to the single-hop experiments, we compare two mapping schemes from content names to faces at the consumer node and subsequent intermediate nodes: (i) a direct assignment of the next-hop MAC address to the corresponding face on the path to the producer, and (ii) a common prefix route where the corresponding face is mapped to a broadcast MAC address.

Even though the same set of nodes is used for (i) and (ii) we cannot guarantee that the same topology appears within the broadcast scenario, as discussed earlier. We do not consider this as a drawback but rather as an advantage, reflecting real-world properties.

4.3.2 Results. In Figure 5, we show the impact of broadcast and unicast faces in a multi-hop network in terms system wakeups and CPU times for a fixed size network and a predefined number of content items per producer. We find similar effects compared to the single-hop scenario, where resource costs for the broadcast mapping (with common prefix routes applied on all nodes) are orders of magnitude higher than for the unicast mapping. The medians of both wakeups and CPU times correspond to our single-hop measurements but larger errors and outliers are visible. The reason for the outlier is rooted in intermediate nodes. These nodes only forward Interest and data packets on the path between producer and consumer. In our measurements, we observed that single links which were stable during the monitoring phase, exhibit asymmetric link behavior later. That led to packet loss of approximately 10% in the unicast setup, whereas the broadcast approach delivered 100% of the requested content items due redundant paths. The resource improvement of name to unicast address mapping as well as the additional MAC layer features such retransmission handling come at the cost of a route maintenance mechanism that is needed to provide fresh and stable links. Analyzing this in more detail, should be part of future work.

5 DISCUSSIONS AND OUTLOOK

In this paper, we discussed current solutions to implement interaction between the NDN network layer and the underlying data link layer. In contrast to the IP network stack, which maps IP addresses to unicast MAC addresses to prevent arbitrary broadcast, there is no such mechanism by default in NDN. Without sacrificing the principle concepts of NDN, we argue that link layer broadcast should be reduced in specific deployment scenarios (e.g., IoT), as it conflicts with limited hardware resources in terms of processing, memory, and energy. We reviewed the current solution space and contributed a first set of experiments in a real testbed. We linked NDN faces to unicast or broadcast MAC addresses and quantified the resource overhead and the advantage of using link layer functions (e.g., retransmission handling).

The position of this paper is threefold. First, an name to link layer mapping is needed and still an open research question. Second, our community should find a solution that does not affect the core of current link layer implementations, and benefits from built-in link layer functions. Third, one promising solution, the dynamic creation of NDN faces, has been mostly ignored and deserves more detailed study. In this paper, we contributed a first set of experiments in a real testbed and related analysis. We hope that the ICN community will find a common understanding of appropriate interaction between network and data link layer in the future.
A Note on Reproducibility
We explicitly support reproducible research [3, 16]. Our experiments have been conducted in an open testbed. The source code of our implementations (including scripts to setup the experiments, RIOT measurement apps etc.) are available on Github at https://github.com/inetrg/ACM-ICN-2017.

Acknowledgments
We thank our shepherd Lixia Zhang and the anonymous ICN reviewers for their valuable feedback. This work was supported in parts by the German Federal Ministry of Research and Education within the project I3: Information-centric Networking for the Industrial Internet.

REFERENCES