Expressive Feature-oriented Multicast for the Internet of Things

Jonathan Oostvogels, Stefanos Peros, Stéphane Delbruel, Danny Hughes

Expressive Feature-oriented Multicast for the Internet of Things

Jonathan Oostvogels, Stefanos Peros, Stéphane Delbruel, Danny Hughes
imec-DistriNet, KU Leuven
3001 Leuven, Belgium
{firstname.lastname}@cs.kuleuven.be

***Abstract*—Applications for the Internet of Things (IoT) are often data-centric. Data-centric routing then enables messages to reach relevant consumers while avoiding flooding and explicit resource discovery. This reduces the amount of transmissions required to support relevant applications and provides energy savings as well as a convenient programming abstraction: messages can be addressed to nodes that advertise features matching a constraint. In low-power wireless mesh networks, such feature-oriented routing traditionally relies on costly and inflexible network overlays. Recent work establishes lightweight support for diverse data-centric traffic patterns, but sacrifices expressiveness of feature-oriented functionality and hence applicability. This paper overcomes that trade-off by introducing SMRFET, a multicasting system that integrates data-centric functionality into a standard low-power network stack. SMRFET thus improves over the art: it offers more expressive addressing mechanics (range queries over a node's features) at lower implementation and runtime cost (no additional networking mechanisms). Additionally, SMRFET can be configured to handle memory constraints: its performance degrades gracefully as the designated amount of memory decreases. SMRFET therefore brings lightweight and expressive group communication to wireless IoT networks.**

***Index Terms*—data-centric routing, multicast, group communication, Internet of Things, RPL, SMRF**

I. Introduction

In low-power wireless mesh networks found within the Internet of Things (IoT), data-centric routing can provide significant energy savings: it provides a communication mechanism that considers application-level concerns instead of numeric addresses, thus avoiding the energy overhead associated with performing network-wide broadcasts or maintaining directory services that map application data to said addresses [1]. The systems described in [2] and [3] implement such data-centric routing by having nodes advertise features. Applications then address messages using a feature constraint: messages reach all nodes that advertise features satisfying the constraint. RFC 7390 [4] illustrates the relevance of this approach by providing an example: building automation using CoAP, where a message has to be sent to all actuators of a given type on a particular floor of a building. A feature-oriented routing system routes messages to the appropriate nodes, i.e. the ones that match the constraints on type, floor and building.

As multiple nodes can satisfy a feature constraint, feature-oriented routing is a form of group communication. In wireless sensor networks and other low-power environments, implementing group communication over multicast offers significant energy savings relative to alternative approaches, such as broadcast and repeated unicast, as the latter techniques relay or duplicate messages unnecessarily [5]. Implementing feature-oriented multicast is challenging, however, as nodes have to communicate, process and store feature information of other nodes to make routing decisions, while being resource-constrained with regard to memory space and transmission time.

To address these challenges, this paper proposes *Stateless Multicast RPL Forwarding with Expressive Targeting (SMRFET)*, a novel feature-oriented multicast system for constrained wireless mesh networks, offering data-centric routing that is more expressive and lightweight than the state of the art. Section II provides an overview of related work and highlights the gap that SMRFET addresses. Next, section III details SMRFET's design, discussing how a standard IoT network stack can support multicast messages that are addressed using range constraints over numeric features advertised by nodes. Finally, Section IV presents this paper's conclusions and outlook.

II. Related work

Unlike e.g. geographic routing, feature-oriented systems cannot exploit a link between a node's connectivity and the properties it advertises, thus requiring explicit routes and storage of the feature information of remote nodes. Many systems ease this burden by limiting themselves to publish-subscribe communication. With *Directed Diffusion (DD)* [1], a seminal example, nodes broadcast advertisements, defining the events they are interested in using a custom key-value based grammar that expresses feature constraints. A reinforcement scheme then gradually establishes multicast routes that deliver event notifications to interested recipients. The authors of [6] use IPv6 multicast to implement a similar form of publish-subscribe, in which multicast addresses encode information about the features to which the subscription relates. Both systems set up additional routing constructs for each data-centric path, i.e. overlay routes or groups for each event.

Several use cases depend on more flexible and unpredictable traffic flows, involving any-to-any, on-demand, in-network communication [7]. Due to memory constraints, naive routing table implementations quickly become prohibitively expensive [7]. Bloom filters [8] are a popular solution: these data structures can summarise a list of descendants along a tree-like topology within a fixed amount of memory. Similar to e.g. *ORPL* [7] for address-centric routing, *Featurecast* [2] uses Bloom filters for feature-oriented communication, thus offering lightweight support for diverse data-centric traffic flows. The system encodes the filters in IPv6 addresses and consults hash-based routing tables to multicast messages along an acyclic graph. This scheme is shown to enable lightweight data-centric communication by comparing its messaging and memory requirements to those of *Logical Neighborhoods (LN)* [3], a system that relies on a custom addressing grammar and local flooding of advertisements. Unlike DD and LN, however, Featurecast's hash-based design does not allow to consider ranges of values: "measuring 25 °C" and "measuring 26 °C" are addressed as entirely different, binary features. Applications cannot benefit from the similarity between both for expressive addressing, nor can it be exploited for memory-efficient routing (cf. [9]). Featurecast's lightweight approach thus reduces expressiveness and applicability.

This paper makes a twofold contribution relative to the state of the art. First, SMRFET reconciles lightweight support for flexible traffic patterns with the level of application-level expressiveness offered by more heavyweight publish-subscribe systems. With SMRFET, multicast messages are addressed using a conjunction of range constraints over numeric features. Secondly, this paper shows how data-centric functionality can be implemented compactly on top of *IPv6 Stateless Multicast RPL Forwarding (SMRF)* [10], a state-of-the art protocol for group-based multicast in low-power networks. While network-layer integration has been considered before [2], [6], no system discussed above supports data-centric communication without additional networking mechanisms.

III. Design

A. Routing

SMRF, and hence SMRFET, rely on *RPL* [11], the IETF standard for unicast in low-power networks. RPL connects nodes along *destination-oriented directed acyclic graphs (DODAGs)*. Each node in a DODAG, except for an arbitrarily chosen root node, is associated with a single *preferred parent*. RPL thus constructs a tree connecting all nodes in a network, with preferred parent-child relations as edges. SMRF implements multicast by broadcasting along that tree, while pruning forwarding branches as messages propagate [10]. The root starts the dissemination of a multicast message with a link-layer broadcast of that message. A node that receives the transmission first checks whether the sender (i.e. the root) is its preferred parent. If not, it ignores the message. Next, the node checks whether it is a member of the multicast group to which the message is addressed. If so, the message is delivered to upper layers in the network stack. Finally, SMRF decides whether to forward (i.e. link-layer rebroadcast) the message based on group subscriptions of its descendants in the tree.

SMRFET differs from SMRF in two aspects. First, SMRFET routes multicast messages based on feature information. To decide whether to forward a message, a node consults a summary of the feature information advertised by descendants instead of a collection of group subscriptions. This forwarding decision, and the routing table that supports it, is detailed in section III-D. SMRFET also addresses messages using feature information instead of multicast groups: sections III-B and III-C discuss how to use regular IPv6 addresses to encode features and constraints. The system can leverage all other routing-related functionality (e.g. tree maintenance after failures) from underlying RPL and SMRF implementations.

B. Advertising features

Inspired by Featurecast [2] and the system proposed in [6], SMRFET encodes feature information in IPv6 multicast addresses. These addresses end in an arbitrary 112-bit group identifier [12], which SMRFET considers as seven two-byte pairs. The first byte identifies a feature: it is the hash of a feature identifier (a string) to $[1, 255]$. A zero byte indicates the absence of a feature. The second byte corresponds to the feature value for that identifier. Using regular IPv6 multicast addresses, nodes thus advertise *feature tuples*, which are in essence collections of key-value pairs. Since the chance of hash collisions is small (3.9% for 5 features), collision handling is left to the application layer.

SMRFET, like SMRF, relies on RPL to disseminate advertisements. Nodes inform their preferred parent of the multicast addresses to which they subscribe by sending it *Destination Advertisement Objects (DAOs)*. DAOs are custom ICMPv6 messages that contain the (multicast) addresses of the node that sends them. Parents in turn forward this information to their parents, propagating the advertisement towards the root. Each node along the forwarding path stores the multicast addresses that are being advertised: it needs this information to decide whether to forward messages.

C. Addressing

Addressing works similarly to advertising: SMRFET encodes feature constraints as IPv6 multicast addresses. The system considers the first twelve bytes of the group identifier as four three-byte fields. The first byte in every field is the hash of a feature identifier, while the second and third specify the minimum and maximum value that recipients should advertise for that feature. SMRFET thus allows to address multicast messages using a conjunction of up to four range constraints on a node's features. The system currently does not rely on the final two bytes of an IPv6 multicast address to specify which nodes should receive a message: applications can use them for arbitrary purposes, such as interoperability with other routing schemes. Messages can be sent over any protocol on top of IPv6: nodes only need to know how to decode addresses and how to make forwarding decisions.

In a standard stack, 6LoWPAN header compression (HC) [13] exploits redundancies in addressing information to reduce packet size. IPv6 addresses that encode feature information, however, cannot be compressed effectively. This protocol interaction increases SMRFET's packet sizes as compared to SMRF, but is transparent from an implementation point of view: SMRFET does not consider HC.

D. Routing table

SMRFET consults a feature-based routing table for forwarding decisions. Naively storing all features can quickly get prohibitively expensive: if 100 nodes each advertise 10 features, the root node needs about 1 kB of RAM to keep track of all information. *Class 1 devices* [14] are limited to around 10 kB of RAM, so this cost is significant.

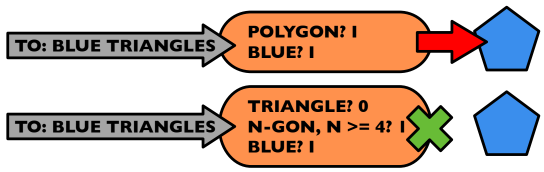

Fig. 1. Bucket-based forwarding: nodes that maintain more data make more accurate forwarding decisions, thus wasting less energy on superfluous transmissions.

SMRFET's routing table is designed to operate within a configurable amount of memory. It consists of an arbitrary amount of *buckets* that summarise feature data. Each bucket is essentially a single bit, initialised to 0, that is associated with a set of feature tuples that may or may not have been advertised. In general, a bucket is associated with multiple tuples and vice versa. When an advertisement arrives, SMRFET relies on a mapping scheme to determine the buckets that are associated with the advertised tuple and sets the corresponding bits to 1. To decide whether to forward a message, a node iterates over all buckets associated with tuples that can satisfy the constraint with which the message has been addressed. If, for none of the tuples, the associated bits are all set to 1, a node drops the message: no descendant has advertised relevant features. Figure 1 illustrates how bucket-based data-centric forwarding introduces a trade-off between memory and radio utilisation.

Such summarisation leads to *false positives*: SMRFET cannot distinguish feature tuples that map to the same bucket. Nodes can falsely appear to satisfy a constraint by advertising tuples registered in the same buckets as tuples that truly satisfy it. Consequently, SMRFET may unnecessarily forward messages and thus incur an energy overhead relative to SMRF. This does not impact applications: SMRFET's addressing logic (cf. section III-C) decides on up-stack delivery. The challenge is to design a scheme that determines the mapping between feature tuples and buckets, using as few buckets as possible while maintaining a low false positive rate. Such a scheme also adjusts the number of buckets to only use available memory, and adapts itself to exploit similarities between features.

SMRFET currently implements *adaptive range filters (ARFs)* [15] to manage its buckets. ARFs, originally proposed in a database context, divide a range of values in dynamically sized buckets: they learn the distribution of advertised values and adjust bucket size, which is the number of values associated with a bucket, to accurately model that distribution. Sensor readings, for example, may cluster around an average value. ARFs then maintain many small buckets for values near that average and use fewer large buckets to cover less densely occupied subranges in order to maintain a low false positive rate. ARFs also adapt their structure to make their (meta)data fit an arbitrary amount of memory: they merge (split) buckets to decrease (increase) memory requirements. The more memory is used to model a range of values, the lower the false positive rate for that range [15]. This approach is a generalisation of Bloom filter based forwarding decisions as described in [2]: Bloom filter routing does not iterate over buckets to allow complex (e.g. range) queries, and hence does not consider how to exploit similarities between feature values to enable expressive functionality in a constrained environment (i.e. with few buckets and iterations).

IV. Conclusion and Outlook

This paper introduces SMRFET, a multicasting system for low-power wireless mesh networks. Using SMRFET, nodes advertise byte-valued features. Applications then address multicast messages with range constraints on those features. Barring link-layer failures, the system efficiently delivers the messages to all nodes that satisfy the constraints. SMRFET relies on the ARF data structure to limit unnecessary forwarding; the system's communication requirements increase gracefully as the amount of RAM allocated to its ARFs decreases.

Preliminary simulation-based experiments, not detailed in this paper, confirm SMRFET's feasbility. The mechanisms in section III require only minor modifications to existing RPL implementations for class 1 devices, and lead to negligible energy overhead relative to traditional group-based multicast at a routing state memory cost of a few bytes per feature. Future work must verify this performance characterisation in real-word deployments and investigate SMRFET's behaviour for feature updates and node mobility. Tailoring SMRFET's generic bucket-based forwarding scheme to specific applications also appears to be a promising area of research.

V. Acknowledgements

This research is partially funded by the Research Fund KU Leuven and the FWO D3-CPS project. The authors thank Nicolas Small for his feedback early in this research.

References

[1] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: a scalable and robust communication paradigm for sensor networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, pp. 56–67, ACM, 2000.
[2] M. Król, F. Rousseau, and A. Duda, "Featurecast: lightweight data-centric communications for wireless sensor networks," in *European Conference on Wireless Sensor Networks*, pp. 202–217, Springer, 2015.
[3] L. Mottola and G. P. Picco, "Logical Neighborhoods: a programming abstraction for wireless sensor networks," in *DCOSS*, vol. 4026, pp. 150–168, Springer, 2006.
[4] A. Rahman and E. Dijk, "Group communication for the Constrained Application Protocol (CoAP)," RFC 7390, RFC Editor, October 2014.
[5] J. S. Silva, T. Camilo, P. Pinto, R. Ruivo, A. Rodrigues, F. Gaudêncio, and F. Boavida, "Multicast and IP multicast support in wireless sensor networks.," *JNW*, vol. 3, no. 3, pp. 19–26, 2008.
[6] S. Akkermans, R. Bachiller, N. Matthys, W. Joosen, D. Hughes, and M. Vučinić, "Towards efficient publish-subscribe middleware in the IoT with IPv6 multicast," in *2016 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2016.
[7] S. Duquennoy, O. Landsiedel, and T. Voigt, "Let the tree bloom: scalable opportunistic routing with ORPL," in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, p. 2, ACM, 2013.
[8] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.
[9] Z. Jerzak and C. Fetzer, "Bloom filter based routing for content-based publish/subscribe," in *Proceedings of the second international conference on Distributed event-based systems*, pp. 71–81, ACM, 2008.
[10] G. Oikonomou, I. Phillips, and T. Tryfonas, "IPv6 multicast forwarding in RPL-based wireless sensor networks," *Wireless personal communications*, vol. 73, no. 3, pp. 1089–1116, 2013.
[11] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, "RPL: IPv6 routing protocol for low-power and lossy networks," RFC 6550, RFC Editor, March 2012.
[12] R. Hinden and S. Deering, "IP version 6 addressing architecture," RFC 4291, RFC Editor, February 2006.
[13] J. Hui and P. Thubert, "Compression format for IPv6 datagrams over IEEE 802.15.4-based networks," RFC 6282, RFC Editor, September 2011. http://www.rfc-editor.org/rfc/rfc6282.txt.
[14] C. Bormann, M. Ersue, and A. Keranen, "Terminology for constrained-node networks," RFC 7228, RFC Editor, May 2014.
[15] A.-K. Alexiou, D. Kossmann, and P.-Å. Larson, "Adaptive range filters for cold data: avoiding trips to Siberia," *Proceedings of the VLDB Endowment*, vol. 6, no. 14, pp. 1714–1725, 2013.