

ML-based Network Pruning for Routing Data Overhead Reduction in Wireless Sensor Networks

Original

ML-based Network Pruning for Routing Data Overhead Reduction in Wireless Sensor Networks / Andreoletti, Davide; Rottondi, Cristina; Ezzeddine, Fatima; Ayoub, Omran; Giordano, Silvia. - ELETTRONICO. - (2023), pp. 122-125. (18th Wireless On-Demand Network Systems and Services Conference (WONS) Madonna di Campiglio (IT) January 30 - February 1, 2023) [10.23919/WONS57325.2023.10061999].

Availability:

This version is available at: 11583/2978618 since: 2023-05-18T19:01:01Z

Publisher:

IEEE

Published

DOI:10.23919/WONS57325.2023.10061999

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

IEEE postprint/Author's Accepted Manuscript

©2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

ML-based Network Pruning for Routing Data Overhead Reduction in Wireless Sensor Networks

Davide Andreoletti¹, Cristina Rottondi², Fatima Ezzeddine^{1,3}, Omran Ayoub¹, and Silvia Giordano¹

¹University of Applied Sciences and Arts of Southern Switzerland, Lugano, Switzerland, Email: {name.surname}@supsi.ch

²Department of Electronics and Telecommunications, Politecnico di Torino, Italy, Email: {name.surname}@polito.it

³Università della Svizzera Italiana, Lugano, Switzerland, Email: {name.surname}@usi.ch

Abstract—Routing in Wireless Sensor Networks (WSNs) is one of the tasks that heavily impact network lifetime: current routing protocols, such as Ad-hoc On-demand Distance Vector (AODV), generate excessive and rather unnecessary overhead for route discovery, which in turn contributes to deplete the limited power resources of sensors. In this work, we propose a novel machine learning-based approach to perform network pruning during route discovery aiming at reducing data overhead. Our approach assumes that sensor nodes are aware of their locations and have processing capabilities to run lightweight machine learning algorithms. We perform extensive simulations considering WSNs consisting of different amounts of nodes. Results show that our proposed approach can reduce data overhead by 50% to 65%, depending on the amount of nodes and pruning aggressiveness.

I. INTRODUCTION

Wireless Sensors Networks (WSNs) consist of a distributed set of autonomous interconnected sensor nodes, which can range from hundreds to thousands. These sensor nodes are primarily used for sensing and monitoring in various applications such as, e.g., industrial factories, environment surveillance and health monitoring, and have stringent battery, storage and processing constraints. In fact, WSNs generally lack an infrastructure for energy supply, and sensors are generally tiny battery-powered devices with a low transmission power. As it is impractical to frequently replace sensor batteries, it is of utmost importance to design efficient strategies to perform main sensor tasks while prolonging batteries life-time.

Packet routing, which is the process of discovering routes to interconnect nodes, represents one of the crucial tasks in WSNs. An efficient routing strategy needs to take into account the specific characteristics of the network, such as topology, mobility and security requirements. Specifically, in WSNs, a desired property of any routing strategy is the minimization of the control data overhead that nodes exchange with one another to establish and maintain routes. In this respect, a broad distinction exists between proactive and reactive protocols. In the former, e.g., Open Shortest Path First (OSPF) [1], the nodes of the network regularly exchange control messages to keep the discovered routes up to date. In the latter, e.g., Ad-hoc On-demand Distance Vector (AODV) [2], the route discovery phase is initiated only when two nodes need to communicate. By triggering the exchange of control messages

only on demand, reactive protocols generate lower overhead with respect to the proactive ones. Further reducing the overhead generated by reactive protocols is crucial to save energy, and therefore prolong network lifetime, especially in large networks composed of a high number of nodes, where data overhead can increase dramatically. For instance, in the AODV protocol, route requests (RREQ) messages are broadcast in the network, and may eventually reach nodes that are located far away from the target destination, which yields unnecessary control signaling overhead.

In this work, we propose a modified version of the AODV protocol that exploits machine learning (ML) strategies to reduce the overhead generated in the route discovery phase by performing network pruning. The implementation of our solution is publicly available at¹. We train an ML algorithm that, given the euclidean distances between a node and a source/destination pair, infers the probability of that node belonging to the path that interconnects them. Based on the outcome of the ML model, a decision whether to keep or discard the node in the discovery process is taken. More specifically, upon receiving a RREQ message for establishing a route between a source and a destination, each node infers the probability of being part of a path between them and, if the probability is lower than a predefined threshold value, referred to as *pruning threshold*, the node does not broadcast the RREQ messages to its neighbors, thus limiting their spread and therefore reducing overhead. As, nowadays, microprocessors for sensors can afford executing light-weight ML algorithms without requiring additional infrastructures [3], pruning can be implemented directly by nodes. We also assume that each node knows its position (e.g., through the GPS), as this information is usually required for various network applications [4]. Moreover, we consider that the position of the source can be included into the RREQ messages, while the position of the destination (which is, in most of the cases, the sink that connects the WSN with an external infrastructure) can be easily propagated into the network (e.g., using the method proposed in Ref. [5]). Hence, nodes can compute their distances with respect to source and destination, and execute

¹<https://github.com/FatimaEzzeddine/ML-based-Network-Pruning-for-Routing-Data-Overhead-Reduction-in-Wireless-Sensor-Networks>

the proposed ML algorithm.

To evaluate the effectiveness of our proposed approach, we perform extensive simulations considering various network topologies and quantitatively compare the overhead (measured as the total number of RREQ messages generated in the route discovery phase) of our approach to that of the AODV protocol. Results show that our proposed ML-based approach for network pruning in WSN can achieve a reduction of up to 65%, while negligibly impacting other routing parameters such as average route length.

The rest of the paper is organized as follows. In Sec. II we briefly review existing literature related to routing in WSN, with special attention to routing strategies enhanced by means of ML techniques. Sec. III describes the proposed ML-based methodology and the modified AODV protocol. Sec. IV presents numerical results, discusses limitation and future research directions. Finally, Sec. V concludes the paper.

II. RELATED WORK

A large variety of routing protocols has been proposed to address the specific requirements of various types of WSNs, such as high mobility [6], security constraints [7] and characteristics of the topology [8] (we refer the reader to Ref. [9] for a detailed survey). Similarly to our work, Ref. [4] proposes a routing protocol for WSN that exploits the knowledge of the position of the destination. In that work, the next node in the route discovery process is selected based on simple hand-crafted rules (i.e., lowest distance with respect to the destination). Instead, we employ ML methodologies to establish which nodes should be excluded (i.e., pruned) from the route discovery process. Other works have proposed enhancements of well-known protocols, e.g., to increase the energy efficiency of the OSPF protocol [10], or to reduce data overhead in the AODV protocol [11]. In particular, similar to our work, Ref. [11] defines a probability for each node to broadcast RREQ messages which depends on the number of its neighbors. Instead, we apply an ML-based network pruning approach to decide whether a node should broadcast RREQ messages. Recently, ML techniques have been extensively applied to WSN routing protocols (see Ref. [12] for a survey). Among these works, ML has been used to enhance AODV only in [13], where a genetic algorithm is employed to select the best among a set of computed paths. To our knowledge, no previous works have focused on enhancing the route discovery phase of AODV, neither on applying network pruning to reduce the overhead of RREQ messages in a WSN network employing AODV. Our application of ML focuses is inspired by our previous work [14], where we applied network pruning to enhance routing in optical networks. In this work, we adapt the methodology proposed in [14] to the AODV protocol (e.g., we define a data representation suitable for WSNs).

III. MACHINE-LEARNING-BASED AD-HOC ON-DEMAND DISTANCE VECTOR

This section describes the design of the ML model used to compute the likelihood that each node belongs to the path between a source/destination pairs, and elaborates on how to integrate this model into the AODV protocol.

A. Machine Learning model

1) *Data Representation:* We observe that nodes belonging to paths discovered by the AODV protocol have comparable topological and geographical characteristics (e.g., similar distances with respect to a set of predefined nodes). Therefore, we propose to represent each node in the network n_i as $[f_s^{(i)}, f_d^{(i)}]$, where $f_s^{(i)}$ and $f_d^{(i)}$ represent the euclidean distances of the i -th node with reference to the source s and to the destination d , respectively. By using this representation strategy, we aim to make nodes belonging to the path between s and d distinguishable from those that do not belong to it. Moreover, the selected representation can be obtained at low processing complexity, and therefore meets the computational capacity of resource-constrained sensors.

2) *Training:* The procedure for generating the dataset for model training is as follows:

- 1) Generate K_{tr} networks, each composed of N_{tr} nodes.
- 2) Randomly select M_{tr} pairs of sources and destinations among the nodes of the network.
- 3) Define two empty lists \mathcal{D}_x and \mathcal{D}_y to store nodes representations and corresponding labels, respectively.
- 4) Execute the AODV protocol to obtain a path between each of the M_{tr} source/destination pairs.
- 5) For every node of the discovered paths (e.g., n_i), add its representation $[f_s^{(i)}, f_d^{(i)}]$ to the list \mathcal{D}_x and the value 1 to the list \mathcal{D}_y .
- 6) Randomly select nodes (e.g., n_j) from the set of nodes that do not belong to the path and add its representation $[f_s^{(j)}, f_d^{(j)}]$ to the list \mathcal{D}_x and the value 0 to the list \mathcal{D}_y .

Using the obtained dataset, an Extreme Gradient Boosting (XGB) model is trained to learn a map between the representation of n_i (i.e., $[f_s^{(i)}, f_d^{(i)}]$) and a ground truth label that is 1 if n_i belongs to the path between source s and destination d , and 0 otherwise. The training is framed as a regression, in such a way that the trained model outputs a probability $p_{sd}^i \in [0, 1]$, which is the likelihood that node n_i belongs to the path between s and d . In order to avoid training a model that is biased towards one of the two classes (i.e., either belonging or not to the path), the generated dataset is balanced by construction, i.e., to have the same number of samples for each class.

B. ML-assisted AODV

The AODV protocol uses two main control messages to perform the route discovery phase, namely Route Requests (RREQ) and Route Reply (RREP). When a source s wants to

communicate with a destination d , it starts broadcasting RREQ to its neighbors, which in turn broadcast the RREQ to their neighbors, and so on until the destination is reached. Upon receiving an RREQ message, the destination forwards back the RREP message, and the communication between s and d can start over the discovered route. Following Ref. [15], we assume that no caching mechanism is set in place. In future work, we will assume that also intermediate nodes which know how to reach the destination can directly forward RREP messages back to the source.

We observe that a RREQ contains the identifier of both source s and destination d . In our implementation, RREQ packets are modified to contain the locations of s and d as well. Upon the reception of a RREQ message, each node (say n_i) computes the distance between its own location and the locations of sources and destination, i.e., it obtains the representation $[f_s^{(i)}, f_d^{(i)}]$, and provides them as input to the ML model. From this query, it obtains the probability $p_{sd}^i \in [0, 1]$, which captures the likelihood that node i belongs to the path between s and d . This probability is then compared with a pruning threshold parameter γ and, if $p_{sd}^i \geq \gamma$, node n_i broadcasts the RREQ message to its neighbors; otherwise, n_i does not forward the RREQ message.

IV. NUMERICAL RESULTS

A. Simulation Settings

We consider an area of 4900 m^2 (70 m length and 70 m wide) with N nodes placed following a uniform distribution within the area. We perform simulations considering different values of N varying from 100 to 1000 with a step of 100. For each value of N , we perform 20 simulations randomly varying the location of the nodes within the network. Following Ref [16], we consider that each node (i.e., each sensor) can reach any other node within a 10-meter range, based on its maximum power transmission range, thus all nodes that fall within a range of 10-meter from one another are considered neighbors. For each simulation, we consider 50 requests generated randomly between source and destination sensors.

We test our proposed approach using a model trained on data collected from $K_{tr} = 50$ networks, each composed of $N_{tr} = 100$ nodes deployed over a 4900 m^2 total surface area (70 m length and 70 m wide). In each of these networks, $M_{tr} = 100$ routes have been computed. We consider three values of the pruning threshold $\gamma \in [0.4, 0.5, 0.6]$. The pruning threshold reflects the aggressiveness of the model: increasing γ decreases the probability that a node belongs to the path between source and destination, and therefore decreases the probability of propagating RREQ messages. On the other hand, decreasing γ increases the probability of propagating RREQ messages.

We measure the effectiveness of our proposed approach by means of the following three metrics:

- Percentage of Paths Found (% *Success*), which measures the percentage of successful path discoveries using the

ML-aided AODV protocol. In fact, pruning the network may disconnect source and destinations, which results in the failure of the route discovery phase.

- Percentage Gain in Overhead (% *Overhead Gain* \mathcal{G}), which is obtained as $\mathcal{G}_{gain} = 100 \cdot \frac{\mathcal{O}_{np} - \mathcal{O}_p}{\mathcal{O}_{np}}$, where \mathcal{O}_{np} is the overhead obtained using the AODV protocol to establish routes, and \mathcal{O}_p is the overhead obtained with our proposed ML-aided AODV. Note that, in case the latter cannot find a path (i.e., because pruning has disconnected the network), the original AODV protocol is executed in the network. The term \mathcal{O}_p accounts for both the overhead generated by the first attempt (with the ML-aided AODV) and with the second route discovery (performed by AODV).
- Percentage of Extra Path Length (% *Extra Path*), which is the percentage extra-length of the paths discovered by the ML-aided AODV with respect to AODV.

B. Discussion

Figure 1 shows the % *Success* for varying N and for different values of pruning threshold γ . First, we note that for $N = 100$ the proposed approach achieves very low % *Success*, ranging between 47% (for a pruning threshold of 0.6) to 70% (for a pruning threshold of 0.4). As N increases to 200, % *Success* increases to 65%, 82% and 90% for a pruning threshold of 0.6, 0.5 and 0.4, respectively, and then % *Success* continues to increase progressively for higher N to reach 75%, 90% and 98% for $N = 1000$ and pruning threshold of 0.6, 0.5 and 0.4, respectively. This shows that our proposed strategy can achieve a near-100 % *Success* for large N assuming a suitable pruning threshold is used. It follows that the pruning threshold should be tuned based on the size of network.

We now discuss the impact of pruning on overhead. Figure 2 shows the % *Overhead Gain* \mathcal{G} for varying N and for the three values of pruning threshold already considered. Results show that for networks with small N , % *Overhead Gain* \mathcal{G} is between 30% and 35%, depending on the pruning threshold used. These values of \mathcal{G} are the lowest seen in our simulations. This is expected as, for $N = 100$, our proposed strategy successfully finds a path between source and destination only in 40% to 70% of the cases, depending on the pruning threshold (see Fig. 1). We recall that, when the ML-aided AODV fails, our proposed strategy incurs extra overhead due to applying the AODV protocol after the failure of the ML-aided AODV, which explains the relatively low % *Overhead Gain* \mathcal{G} . For larger values of N , \mathcal{G} increases to reach 65%, 62% and 55% at $N = 1000$ for pruning threshold $\gamma = 0.6, 0.5$ and 0.4 , respectively. We highlight that, even with an aggressive pruning strategy with a threshold of 0.6, which yields lower % *Success* than that of 0.5 and 0.4, % *Overhead Gain* \mathcal{G} is the highest. This is due to the fact that the overhead generated when the ML-aided AODV protocol can find a path on an aggressively-pruned network is significantly lower than the overhead generated on the original network. This

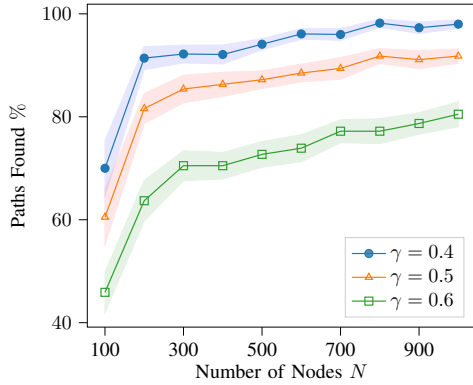


Fig. 1. Percentage of paths found by the ML-aided AODV protocol, for several number of nodes N and pruning threshold γ

well compensates the additional overhead generated by the re-execution of the AODV protocol when a path cannot be found with the ML-aided AODV approach, thus yielding a high % *Overhead Gain*. In terms of length of the discovered lightpaths, the average path length increases by less than 5% in all cases, showing that our proposed approach can significantly reduce the routing overhead without heavily impacting the path length. The following evaluations will be performed in future work:

- Impact on delay: the ML-aided AODV protocol introduces less delay with respect to AODV; however, an additional delay is introduced when the pruning process disconnects source and destination pairs, as a re-execution of AODV on the original network needs to be performed.
- Impact on network lifetime: by reducing the overhead, the ML-aided AODV protocol is also expected to increase network lifetime, because of the reduced power consumption of sensors. However, the power consumed to execute the ML model needs to be taken into consideration.
- Evaluation of the robustness of the ML-aided AODV with respect to various system parameters, such as mobility and transmission power of the nodes.

V. CONCLUSION

In this work, we propose a novel machine learning-based network pruning algorithm and combine it with the AODV algorithm to reduce data overhead in the route discovery phase in WSNs. The ML algorithm infers the probability that a node belongs to the path connecting a given source destination pair and decides, based on a pruning threshold, whether to remove or include the node in the pruned network. A pruned node does not forward RREQ messages to its neighbors. By the means of simulations, we compare the performance of our proposed approach to that of AODV considering different number of nodes in the WSN. Results show that our proposed approach successfully discover a route in up to 90% of the queries and can reduce data overhead by 50% to 65%, depending on the amount of nodes and pruning aggressiveness. As a future

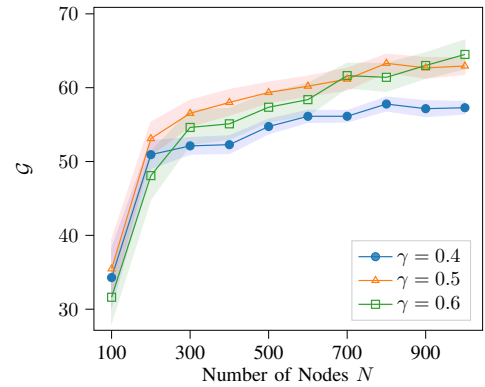


Fig. 2. Percentage gain of overhead \mathcal{G} , for several number of nodes N and pruning threshold γ

work, in addition to optimizing pruning threshold, we plan to consider the impact of our approach on the overall route discovery delay and required power consumption.

REFERENCES

- [1] J. Moy "Rfc2328: Ospf version 2," 1998.
- [2] C. Perkins et al. "Rfc3561: Ad hoc on-demand distance vector (aodv) routing," 2003.
- [3] P. Warden et al. "TinyML." O'Reilly Media, Incorporated, 2019.
- [4] Y. Yu et al. "Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks," 2001.
- [5] F. Yu et al. "A simple location propagation scheme for mobile sink in wireless sensor networks," *IEEE Communications Letters*, vol. 14, no. 4, pp. 321–323, 2010.
- [6] V. G. Menon et al. "Analysing the behaviour and performance of opportunistic routing protocols in highly mobile wireless ad hoc networks," *International Journal of Engineering and Technology*, vol. 8, no. 5, pp. 1916–1924, 2016.
- [7] N. Nasser et al. "Seem: Secure and energy-efficient multipath routing protocol for wireless sensor networks," *Computer communications*, vol. 30, no. 11–12, pp. 2401–2412, 2007.
- [8] G. Dhand et al. "Smeer: Secure multi-tier energy efficient routing protocol for hierarchical wireless sensor networks," *Wireless Personal Communications*, vol. 105, no. 1, pp. 17–35, 2019.
- [9] S. K. Singh et al. "Routing protocols in wireless sensor networks—a survey," *International Journal of Computer Science Engineering Survey (IJCSES)*, vol. 1, no. 2, pp. 63–83, 2010.
- [10] A. Cianfrani et al. "An ospf enhancement for energy saving in ip networks," *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2011, pp. 325–330.
- [11] M. Malnar et al. "An improvement of aodv protocol for the overhead reduction in scalable dynamic wireless ad hoc networks," *Wireless Networks*, vol. 28, no. 3, pp. 1039–1051, 2022.
- [12] P. Nayak et al. "Routing in wireless sensor networks using machine learning techniques: Challenges and opportunities," *Measurement*, vol. 178, p. 108974, 2021.
- [13] V. Trivedi et al. "Modified aodv using genetic algorithm to minimize energy consumption in manet," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, pp. 525–530, 2019.
- [14] D. Andreoletti et al. "A machine learning framework for scalable routing and wavelength assignment in large optical networks," *Optical Fiber Communication Conference*. Optica Publishing Group, 2021, pp. F2G–3.
- [15] J. Thamalakal et al. "Novel caching mechanism to improve performance in mobile ad-hoc networks (manets)," *22nd International Computer Science and Engineering Conference (ICSEC)*. IEEE, 2018, pp. 1–5.
- [16] L.-L. Hung et al. "Energy-efficient cooperative routing scheme for heterogeneous wireless sensor networks," *IEEE Access*, vol. 8, pp. 56 321–56 332, 2020.