Energy-efficient Broadcast Trees for Decentralized Data Dissemination in Wireless Networks

Artur Sterz*†, Robin Klose*‡, Markus Sommer†, Jonas Höchst†, Jakob Link‡, Bernd Simon§, Anja Klein§, Matthias Hollick‡, Bernd Freisleben†

†Dept. of Mathematics & Computer Science, University of Marburg, Germany
E-mail: {sterz, msommer, hoechst, freisleb}@informatik.uni-marburg.de
‡Dept. of Computer Science, Technical University of Darmstadt, Germany
E-mail: {rklose, jlink, mhollick}@seemoo.de
§Dept. of Electrical Engineering & Information Technology, Technical University of Darmstadt, Germany
E-mail: {b.simon, a.klein}@nt.tu-darmstadt.de

2023 IEEE 48th Conference on Local Computer Networks (LCN) | 979-8-3503-0073-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/LCN58197.2023.10223400

***Abstract*—We present a novel multi-hop data dissemination protocol for wireless networks that minimizes the total energy consumption across an entire network by minimizing the transmission power at each hop. It is based on a game-theoretic model, constructs a spanning tree topology in a decentralized manner, and is usable in practice. We evaluate the protocol via simulation and a pratical implementation on a testbed of 75 Raspberry Pis, demonstrating that a total energy reduction of up to 90% can be achieved compared to a simple broadcast protocol.**

***Index Terms*—Wireless network, data dissemination, broadcast tree**

I. Introduction

In wireless networks, disseminating data across multiple devices is required in several application areas, e.g., in wireless sensor networks and the Internet of Things. This involves broadcasting data to all nodes in a network for tasks such as network configuration, update diffusion, and event distribution.

Energy-efficient broadcasting is crucial due to the limited energy budgets of the participating wireless devices, and a multi-hop data dissemination scheme is necessary to reach nodes positioned beyond the transmission range of a sender. A promising approach is to use a spanning tree topology with minimal transmission power at each hop to achieve overall energy efficiency. However, constructing an energy-minimal spanning tree is NP-hard [9].

Several existing spanning tree approaches proposed in the literature are limited in terms of practical applicability, since they either assume global state knowledge or face significant drawbacks in their practical implementation.

We present the Broadcast Tree Protocol (BTP), a novel decentralized approach to construct a spanning tree while minimizing energy consumption for multi-hop data dissemination. BTP is based on a game-theoretic model that is modified to make BTP feasible for a practical implementation.

We evaluate BTP via simulation and deployment on a testbed of 75 Raspberry Pis. Our simulation shows that BTP is on-par with the results of the original game-theoretic model and outperforms approaches from the literature. In our experiments, BTP achieves energy reductions of up to 90% compared to a simple broadcast protocol. To the best of our knowledge, BTP is the first broadcast tree protocol that is based on a proven optimal game-theoretic model and is implemented on real hardware.

The code of our implementation[1], the code for reproducing our experiments[2], and all experimental artifacts[3] are released under a permissive open-source license.

II. Related Work

Compared to BTP, approaches for designing energy-efficient broadcast trees in a decentralized manner published in the literature suffer from various problems. Several approaches require global knowledge of parameters or the states of neighboring nodes [14], [5], [2], [7], [8], [11], [4]. Some decentralized approaches do not consider cycle detection or prevention [2], [11]. Other works do not consider the initial tree construction phase, but assume that an already constructed tree exists where afterwards only the transmission power is adjusted [5], [10], [4], [6], [3]. Furthermore, several approaches give away potential by not leveraging the transmission power of the nodes [2], [10], [1], [4], [6], [3]. Moreover, centralized approaches are usually not suitable in the area of wireless ad-hoc and multi-hop networks [1], [6]. Spanning trees offer the possibility to reach all nodes in a network with the minimally required transmission power, making approaches that do not rely on the tree structure questionable in terms of minimizing energy consumption In contrast to other topologies, [10], [4], [6], [3], spanning trees enable to reach all nodes in a network with the minimum required transmission power. Finally, none of the above mentioned approaches provides an implementation or a practical evaluation using off-the-shelf Wi-Fi devices. In fact, most of the presented works only propose a theoretical model without considering its applicability, making it impossible to implement them under real-world constraints.

*Both authors contributed equally to this research.

[1]https://github.com/umr-ds/broadcast-tree-protocol
[2]https://github.com/umr-ds/broadcast-tree
[3]https://uni-marburg.de/crNaSU

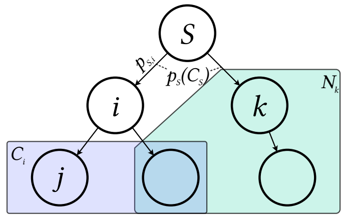

Fig. 1. Broadcast tree overview

III. System Model

We consider a network consisting of multiple nodes $i \in V$ that can receive and transmit data over a wireless channel. The data available at the source node S needs to be disseminated to all nodes $j \in V$ in the network. A multi-hop transmission can be realized using a spanning tree, called *broadcast tree*, as shown in Figure 1, represented by a graph $T = (V, E)$. The source node S is the root, nodes are the vertices of V, and connections with corresponding transmission power weights are the edges in E. The broadcast tree is acyclic and each node $j \in V$ has one parent i except the root S. Parents may have multiple children C_i as shown by the blue box on Figure 1. Using broadcasts, each node i can transmit its data simultaneously to all its children C_i.

To establish a connection between parent i and its children C_i, node i has to send using a transmission power such that for a child $j \in C_i$, the received signal strength must be higher than a certain noise threshold. The signal-to-noise-ratio (SNR) γ_j for receiver j is defined as $\gamma_j = \frac{p_i |h_{i,j}|^2}{\sigma^2}$. p_i is the transmission power of i, $|h_{i,j}|^2$ is the channel gain and σ^2 is the noise.

Furthermore, based on the minimum required SNR γ^{min}, the transmission power from parent i to child j is $p_{i,j} = \frac{\gamma^{min}\sigma^2}{|h_{i,j}|^2}$. In Figure 1, this is represented by the edge between S and i and the value $p_{S,i}$.

Furthermore, a receiving node j can calculate the required transmission power from i to j by $p_{i,j} = \frac{p_i \gamma^{min}}{\gamma_j}$.

The transmission power of parent i is chosen as the maximum transmission power required to reach all of its children: $p_i(C_i) = \max_{j \in C_i}(p_{i,j})$. In Figure 1, these are two connections of S marked with $p_S(C_S)$.

The neighborhood of a parent k is defined as $N_k = \{l | l \in V, p_{k,l} \leq p^{\max}\}$ i.e., all nodes that can be reached by parent k with transmission power $p^{\max}$ or less, visualized by green box in Figure 1.

Our goal is to minimize the total transmission power $p = \sum_{i=1}^{n} p_i(C_i)$ of the network.

IV. Broadcast Tree Protocol

A. Potential Game

Our approach is based on a *potential game*, i.e., all nodes cooperate to minimize the total transmission power required to disseminate the data over the entire broadcast tree T. Using potential games to construct energy-efficient broadcast trees decentralized has been proposed by Mousavi et al. [12]. However, a practical implementation of Mousavi et al.'s potential game faces several challenges. First, the assumption that each node j has knowledge of all potential parents in V not feasible in real-world systems. Second, the discrete time step iterations assumed by the potential game are impractical in distributed algorithms. Third, the potential game uses a weakly dominant strategy. In a practical implementation, this could lead to instability if a child node repeatedly switches between two potential parents with equal transmission power. Fourth, while the potential game assumes the absence of cycles, a practical implementation requires a mechanism to ensure this property.

We propose the following modifications of the potential game to overcome these limitations. Instead of synchronized actions of all nodes simultaneously, each node j decides autonomously whether to switch to a potential parent i or to stay with the current parent Q_j. We implemented a discovery mechanism such that each child node j can discover its potential parents in V. By adopting a strictly dominant strategy, node j switches to potential parent i if its marginal contribution to i is lower than to the current parent Q_j. Since nodes using BTP do not know all potential parents, there is no clear criterion to stop the game. Therefore, we introduce a counter to track the number of decisions without a parent switch. Once a threshold is reached, the node is considered to be finished. Overall, BTP converges to a Nash Equilibrium (NE) similar to the original algorithm of Mousavi et al. [12] and finds the same solution when all neighbors are discovered.

B. BTP

BTP consists of two phases: 1) the broadcast tree construction phase, where the broadcast tree T is constructed, with S as the initializing and root node, and 2) the data dissemination phase, where the actual data is sent from S to all other nodes.

1) Broadcast Tree Construction Phase: To initialize the broadcast tree construction, the source node S sends beacons at maximum transmission power $p^{\max}$. Nodes already in the broadcast tree T also periodically broadcast beacons with $p^{\max}$. Upon receiving a beacon, a receiver j checks if the sender i can be its parent Q. Two cases may arise.

First, if j is not connected to any parent, it requests i to become its child. Second, if j has already a parent Q_j, different from i, it must decide whether switching from Q_j to i would decrease the total transmission power p. This decision depends on four transmission power values: $p_{Q_j,j}$, $p_{Q_j}(C_{Q_j} \setminus j)$, $p_{i,j}$, and $p_i(C_i \cup j)$.

To calculate $p_{Q_j,j}$ and $p_{i,j}$, the transmission power is included in every BTP packet. However, $p_{Q_j}(C_{Q_j} \setminus j)$ and $p_i(C_i \cup j)$ cannot be calculated by j alone. Therefore, these values must be sent with every BTP packet from every node.

Based on the availability of these four values, j switches to i if the following gain vs. loss condition holds: $p_i(C_i \cup j) - p_{i,j} < p_{Q_j,j} - p_{Q_j}(C_{Q_j} \setminus j)$, i.e, if the current parent Q_j can reduce its transmission power more than the potential parent i would need to increase it, j switches parents.

If j decides to connect to the chosen parent i, i verifies that j is not already its parent or its child. If these checks pass, i accepts j as its child or rejects it otherwise. Upon acceptance,

i adjusts its transmission power $p_i(C_i \cup j)$. When i accepts j as its child, j disconnects from its old parent Q_j, if present. Q_j removes j from its child list C_{Q_j} and adjusts its transmission power accordingly to reach the farthest child $k \in C_{Q_j} \setminus j$.

Once a node j does not switch to a new parent a defined number of times, j considers itself finished and notifies its current parent Q_j. Upon receiving this notification from all its children $k \in C_{Q_j}$, Q_j reports its readiness to its parent. The construction phase continues until the source node S receives these notifications from all its children $l \in C_S$. After finishing the construction phase, the data dissemination phase begins.

During tree construction, cycles are detected and resolved. Three scenarios can lead to cycles: when the source node S tries to connect to another node as its parent, when a parent node i attempts to connect to one of its children $j \in C_i$, and when a parent node i tries to connect to a node k that is already on the path from k to S. To handle the last scenario, we propose the Ping-to-Source cycle detection algorithm. Nodes send unicast packets to their parents, and if a packet eventually reaches the node that sent it, a cycle is detected.

2) Data Dissemination Phase: During the data dissemination phase, the source node S transmits data to its children $l \in C_S$ using application data packets with transmission power $p_S(C_S)$. Each node relays the data to their children with their respective transmission power $p_l(C_l)$. The data is split into chunks with increasing sequence numbers, allowing nodes outside the immediate neighborhood to receive and utilize the packets based on the sequence number.

V. Simulation

In this section, our MATLAB simulation of BTP and algorithms from the literature is presented. The number of nodes varies between 10 and 90, and they are randomly placed in a 500 m $\times$ 500 m square area. A random source node is used in each simulation. $p^{\max}$ is set to 20 dBm. For the path loss model, $|h_{i,j}|^2 = \frac{1}{d^\alpha}$ is used. d denotes the distance between nodes i and j, $\alpha = 3$ is the attenuation exponent. $\gamma^{min} = 10$ dB, and the noise power is set to -90 dBm. In addition to BTP, we implemented five baseline algorithms: Dijkstra, BIP [14], BIPSW [14], PCP [7], the original algorithm of Mousavi et al. [12] (BPG), and a variant of BTP that always uses $p^{\max}$ to construct the tree, called Simple Broadcast Protocol (SBP). In total, for every parameter combination, 1,000 runs were simulated, resulting in 10,000 simulations.

In Figure 2, the total transmit power is plotted against the number of nodes in the network. Four performance categories can be identified. SBP performs the worst, deteriorating as more nodes are added. Dijkstra uses less power than SBP, but more than other algorithms. BIP, PCP, BTP, BPG, and BIPSW exhibit better performance. BIPSW is slightly superior with fewer nodes, while BTP and BPG excel with more nodes. However, all algorithms except BTP rely on perfect global knowledge of the network, which is unrealistic. BIP and BIPSW require significant algorithm redesigns, since they rely on communication of states between all nodes, even those out of transmission range. To summarize, our simulation results

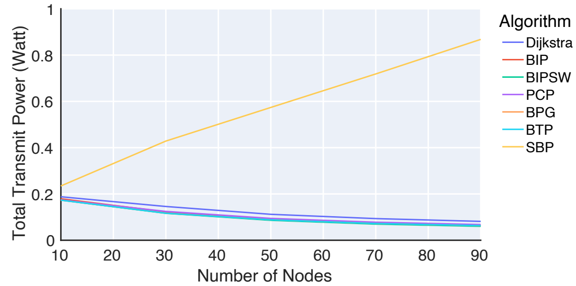

Fig. 2. Total required transmit power for various algorithms

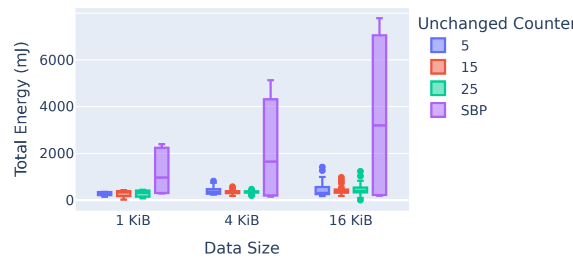

Fig. 3. Total required energy

demonstrate that BTP matches or outperforms other algorithms while operating under realistic assumptions.

VI. Implementation

Our BTP implementation in userland C relies on Wi-Fi as the wireless technology, although BTP is not limited to Wi-Fi. The SNR calculation requires information from the RadioTap header, which is typically unavailable to userland programs. Therefore, we developed a patch using the *Nexmon-framework* [13][4] to preserve the RadioTap header for BTP.

A. Evaluation

To evaluate BTP, we used a testbed consisting of 75 Raspberry Pis across multiple floors of a university building.

Three source nodes were used at different locations in the building: in the northern part of the building, in the center, and in the southern part. Three data sizes were used, 1KiB, 4 KiB and 16 KiB. Unchanged counters of 5, 15, and 25 were used to determine the optimal number of iterations for nodes to consider the broadcast tree construction phase complete. Beside BTP also SBP was evaluated. Each experimental configuration was repeated five times, resulting in a total of 270 experimental runs.

B. Results

a) Total Energy Consumption: Figure 3 illustrates total energy consumption for different parameter sets. Colors indicate different unchanged counters and SBP. BTP achieves an energy reduction of 68-90% compared to SBP, depending on data sizes and unchanged counter values, thanks to its optimized broadcast tree structure. Most notably, while SBP exhibits a 70% energy increase for 16 KiB data compared to 1

[4]https://nexmon.org

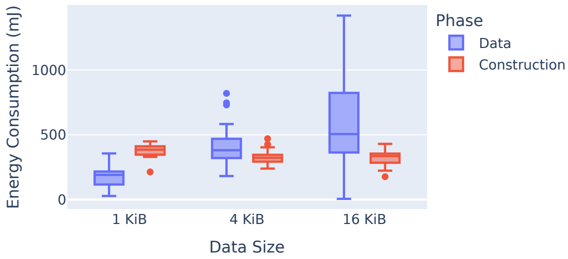

Fig. 4. Energy required for broadcast tree construction and data dissemination

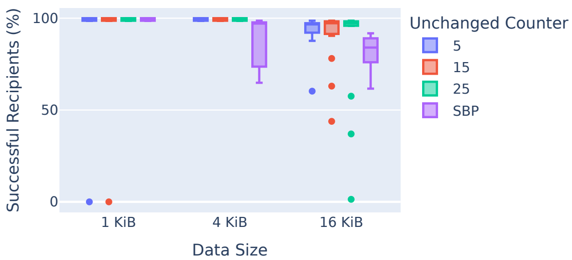

Fig. 5. Percentage of nodes successfully receiving data

KiB, BTP only experiences a 20% increase. This unexpected finding can be attributed to RF interference. Although both protocols lack congestion control mechanisms, BTP's lower power transmission reduces interference, resulting in a moderate overall energy consumption increase.

b) Energy Consumption for Tree Construction and Data Dissemination: In Figure 4, the energy consumption for broadcast tree construction and data dissemination is presented. The energy required for tree construction remains consistent across data sizes, showing no significant differences. Conversely, the energy needed for data dissemination rises with larger data sizes. Notably, even with 1 KiB data, data dissemination accounts for approximately 30% of the energy, and tree construction accounts for the remaining 70%. However, BTP outperforms SBP in terms of total energy consumption, as demonstrated in Figure 3.

c) Successful Receptions: In Figure 5, successful delivery results are depicted. Notably, BTP performs at least as well as SBP, and SBP's performance gets worse with larger data sizes. For 1 KiB and 4 KiB data, BTP achieves nearly 100% delivery, with a few outliers. In contrast, SBP achieves this success ratio only for 1 KiB data, with delivery rates falling below 85% for 4 KiB and 16 KiB sizes. BTP maintains an average delivery rate of approximately 98% for 16 KiB data. Varying network conditions and the wireless medium contribute to these results, since the testbed was deployed in a university building with other Wi-Fi networks and user activities during daytime. BTP's good performance is based on its ability to utilize the wireless medium more efficiently, avoiding network flooding and reducing interference between stations.

VII. Conclusion

The proposed Broadcast Tree Protocol (BTP) is a novel multi-hop data dissemination protocol for wireless networks, designed to minimize energy consumption by constructing a spanning tree in a decentralized manner based on a game-theoretic model. We demonstrated the effectiveness of BTP by simulations and by an implementation on a testbed of 75 Raspberry Pis. Our evaluations showed that BTP can achieve a total energy reduction of up to 90% compared to a simple broadcast protocol.

In future work, we plan to improve BTP by considering node mobility, mechanisms for reliable data transfer, and bi-directional data dissemination.

Acknowledgments

We thank Mahdi Mousavi for making his original MATLAB code [12] available to us. This work is funded by the Hessian State Ministry for Higher Education, Research and the Arts (HMWK) (LOEWE emergenCITY, LOEWE Natur 4.0), and the German Research Foundation (DFG, Project 210487104 - Collaborative Research Center SFB 1053 MAKI).

References

[1] A. Ahmed, S. Abdullah, M. Bukhsh, I. Ahmad, and Z. Mushtaq, "An energy-efficient data aggregation mechanism for iot secured by blockchain," *IEEE Access*, vol. 10, pp. 11 404–11 419, 2022.

[2] L. Cheng, J. Niu, C. Luo, L. Shu, L. Kong, Z. Zhao, and Y. Gu, "Towards minimum-delay and energy-efficient flooding in low-duty-cycle wireless sensor networks," *Computer Networks*, vol. 134, pp. 66–77, 2018.

[3] A. Dusia, R. Ramanathan, W. Ramanathan, C. Servaes, and A. S. Sethi, "Echo: Efficient zero-control-packet broadcasting for mobile ad hoc networks," *Trans. Mob. Comp.*, vol. 21, no. 9, pp. 3163–3175, 2021.

[4] S. Karimi-Bidhendi, J. Guo, and H. Jafarkhani, "Energy-efficient deployment in static and mobile heterogeneous multi-hop wireless sensor networks," *Trans. Wireless Comm.*, vol. 21, pp. 4973–4988, 2021.

[5] M. Khabbazian and K. G. Saffar, "The gain of energy accumulation in multi-hop wireless network broadcast," *Transactions on Networking*, vol. 27, no. 5, pp. 1830–1844, 2019.

[6] M. N. Khan, H. U. Rahman, M. A. Almaiah, M. Z. Khan, A. Khan, M. Raza, M. Al-Zahrani, O. Almomani, and R. Khan, "Improving energy efficiency with content-based adaptive scheduling in wireless sensor networks," *IEEE Access*, vol. 8, pp. 176 495–176 520, 2020.

[7] A. Kuehne, H. Q. Le, M. Mousavi, M. Wichtlhuber, D. Hausheer, and A. Klein, "Power control in wireless broadcast networks using game theory," in *Int. ITG Conf. on Sys., Comm. and Coding*, 2015, pp. 1–5.

[8] S. Kumar, A. Goswami, R. Gupta, S. P. Singh, and A. Lay-Ekuakille, "A game-theoretic approach for cost-effective multicast routing in the internet of things," *IoT*, vol. 9, no. 18, pp. 18 041–18 053, 2022.

[9] D. Li, X. Jia, and H. Liu, "Energy efficient broadcast routing in static ad hoc wireless networks," *Transactions on Mobile Computing*, vol. 3, no. 2, pp. 144–151, 2004.

[10] K. Moon, D.-S. Yoo, W. Lee, and S.-J. Oh, "Receiver cooperation in topology control for wireless ad-hoc networks," *Transactions on Wireless Communications*, vol. 14, no. 4, pp. 1858–1870, 2014.

[11] M. Mousavi, H. Al-Shatri, and A. Klein, "Cost sharing games for energy-efficient multi-hop broadcast in wireless networks," *Transactions on Wireless Communications*, vol. 19, no. 1, pp. 310–324, 2019.

[12] M. Mousavi, H. Al-Shatri, M. Wichtlhuber, D. Hausheer, and A. Klein, "Energy-efficient data dissemination in ad hoc networks: Mechanism design with potential game," in *Int. Symp. on Wireless Communication Systems*. IEEE, 2015, pp. 616–620.

[13] M. Schulz, D. Wegemer, and M. Hollick, "The nexmon firmware analysis and modification framework: Empowering researchers to enhance wi-fi devices," *Computer Communications*, vol. 129, pp. 269–285, 2018.

[14] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Energy-efficient broadcast and multicast trees in wireless networks," *Mobile networks and applications*, vol. 7, no. 6, pp. 481–492, 2002.