Efficient Data Gathering for Decentralized Diversity Combining in Heterogeneous Sensor Networks

Muhammad Nabeel, Vishal Kumar Singh and Falko Dressler
Heinz Nixdorf Institute and Dept. of Computer Science, Paderborn University, Germany
{nabeel,vishal.singh,dressler}@ccs-labs.org

***Abstract*—We study reliable communication in heterogeneous sensor networks. In many application scenarios, more energy-constrained (mobile) nodes are distinguished from more powerful base stations (or ground nodes). Wildlife monitoring is just one of many examples within the Internet of Things research community. In order to improve the communication reliability (and, thus, also the energy footprint), these ground nodes often apply macro-diversity to reduce transmission failures and to avoid costly retransmissions. In recent years, the concept of using distributed sensor networks as antenna arrays for receive diversity has been proposed. We address one of the key challenges in such networks, which is the huge cost of forwarding signal samples to a (central) sink through the ground network, where diversity algorithms are eventually applied. In particular, we present two algorithms, a cluster and a tree based one, that help reducing the data transfers in the ground network. Ground nodes try applying diversity techniques early whenever they receive signal samples from multiple receivers. In extensive simulations, we show that the algorithms substantially outperform naïve centralized solutions.**

I. INTRODUCTION

Wireless Sensor Networks (WSNs) have been successfully used in the past to study and monitor wildlife over a long period of time without any human intervention [1]–[3]. Usually a miniature node (here referred to as a mobile node, i.e., energy and size constrained) is attached to the targeted object which continuously records the required information and transmits it to a stationary ground network upon request [3]. The ground network is composed of distributed sensor nodes (here referred to as ground nodes), which have less strict energy limitations, and are connected to each other and to a centralized sink. If ground nodes detect any transmissions from the mobile nodes, the data is decoded and upon successful decoding forwarded to the sink. Similar architectures can be found in a variety of sensor network applications, where heterogeneous nodes are used for gathering and collecting sensor data. One of the most important challenges is reliable transmission from the mobile nodes to the ground network as retransmissions might be not possible due to connectivity or energy constraints.

The mobile node very likely transmits with a rather low transmit power due to its limited available energy, thus, the transmissions are highly affected by the adverse channel effects such as multi-path fading and Free Space Path Loss (FSPL). Therefore, there is a high probability that the ground nodes which detect the data, fail to correctly decode any information on their own and, hence, nothing is forwarded to the sink. Diversity combining has been proposed to overcome such issues. While normally relying on MIMO systems, receive diversity can also be applied by using the ground network as a distributed antenna array [4]–[6].

The most naïve solution is to forward the detected data from different ground nodes to the sink without decoding. The sink then combines the data copies detected at different nodes constructively. If the channel between different detecting nodes is uncorrelated, such diversity combining improves the received signal strength and helps in successful decoding [5]. However, forwarding multiple data copies from different ground nodes at the same time gives rise to a new challenge due to the network links that offer only limited data rate between nodes.

To cope with this challenge, data aggregation algorithms have been proposed [7]. However, the focus is usually a stationary homogeneous sensor network in which all nodes have similar size and energy limitations or a heterogeneous network where nodes with different capabilities are uniformly distributed. In contrast to the literature, we consider a heterogeneous sensor network which consists of three different types of nodes, i.e., mobile, ground, and sink. The mobile node is highly energy-constrained due to its limited size and may move with high speed. Moreover, it transmits data with a low power whenever in vicinity of any ground node. Ground nodes are comparatively a larger sensor that has less strict energy limitations and are placed statically in the environment. Multiple ground nodes form a stationary network which aims to extract all information reliably from the mobile node to forward it to the sink. Finally, the sink is the most powerful node connected to the edge of ground network for final processing of the received data.

To pass data reliably and efficiently from mobile node to the sink, we propose a variant of both cluster and tree algorithms, and explore the idea of diversity combining for the reduction of data rate requirement in the ground network.

Our main contributions can be summarized as follows:

- We propose variants of cluster and tree algorithms for applying diversity combining locally at ground nodes.
- We consider a wildlife monitoring scenario and develop the whole system model in a simulation environment to investigate the performance.
- We present results from an extensive simulation study, which clearly demonstrate that the proposed variants of cluster and tree algorithms not only reduce the data rate required in the ground network but also improve the network lifetime while achieving the required performance.

II. RELATED WORK

Data gathering or aggregation in sensor networks have been a research focus for many years [7]. Data aggregation reduces redundant information and increases the lifetime of sensor nodes. As a result, the amount of data sent through the network also reduces. Commonly used data aggregation algorithms are classified as cluster-based and tree-based. In cluster-based algorithms [8], [9], clusters are formed within the network and a Cluster Head (CH) is elected in each cluster by a variety of methods. The CH gathers data from its Cluster Members (CMs) and performs data aggregation locally before forwarding the data towards the sink either directly or using intermediate network nodes. In tree-based algorithms [10], [11], a spanning tree rooted at the sink is constructed in which the data trickles down from the leaves towards the root. It means that the data aggregation is performed within the network at intermediate nodes while forwarding the data towards the sink. Tree algorithms are energy-efficient and require only local information about the network topology. To further improve the energy-efficiency in WSNs, hybrid cluster-tree based algorithms also exist [12].

If the transmitting sensor node is highly mobile, data aggregation algorithms can be still successfully applied [13]. However, most of these algorithms target homogeneous sensor networks in which all nodes have similar capabilities and energy limitations. Therefore, to optimize performance of heterogeneous sensor networks, several other algorithms have been proposed such as in [14]. Nevertheless, these works consider that nodes with different capabilities are uniformly distributed within the same area and, hence, use powerful nodes to perform higher processing functions improving the lifetime of energy-constrained sensors. Moreover, the main concern of most algorithms is the energy efficiency of sensor nodes due to their limited capability and available energy.

Here, we consider a heterogeneous sensor network in which the architecture deployed and protocol used are different. Only the transmitting sensor node is very energy-constrained due to which the transmit power is very low. Furthermore, the data that is transmitted to the ground nodes is crucial because the loss of reception can lead to a loss of data forever. In our previous work [6], we proposed to forward detected signal samples that with high probability belong to the same packet from all ground nodes to a central sink for centralized diversity combining. With the proposed technique, the data that needs to be passed in the network reduces dramatically, however, it is still not realistic due to the forwarding of all data to one centralized unit through capacity limited network links.

In this work, we target two research challenges: reducing the data rate requirement in the ground network while applying diversity combining efficiently and achieving a satisfactory reception performance within given time interval. For that, we develop a variant of both cluster and tree algorithms in which diversity combining is applied already early at the ground nodes in a time-efficient manner before forwarding the combined data to the sink.

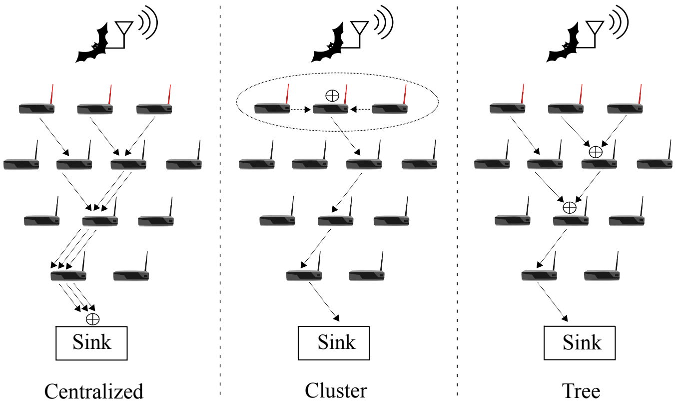


Figure 1. Mobile bat transmitter and distributed ground nodes that detect the transmitted signal. The addition function shows where diversity combining is employed. Arrows indicate necessary transmissions.

III. DATA GATHERING ALGORITHMS

In this section, we introduce both the naïve centralized and two proposed improved solutions using cluster and tree algorithms, respectively. A conceptual overview is shown in Figure 1. It is important to note that we assume routing in our ground network is performed by any ad hoc routing protocol, forming a multi-hop network towards the sink. The centralized approach (cf. Algorithm 1) forwards all detected signal samples to the sink for employing diversity combining at a single point.

In contrast, in the cluster approach (cf. Algorithm 2), ground nodes that detect the transmitted data from the mobile node form a cluster. The involved nodes select and start a *cBackoffTime* timer based on the Signal-to-Noise Ratio (SNR) to decide if they want to become a CH. The ground node with the smallest *cBackoffTime* starts the process, broadcasts a CH selection message, and starts its *slaveBackoff* timer. CMs receiving the CH broadcast forward their signal copy to CH and cancel *cBackoffTime*. Finally, on the expiry of *slaveBackoff*, the CH applies diversity combining locally before forwarding the decoded data to the sink as a single data message. In the algorithm, SNR_{max} and SNR_{min} represent configurable SNR

Algorithm 1 `Centralized`

```
Require: event ∈ {signal from mobile node, signal from
    ground node}
Ensure: received signal is forwarded to sink
switch (event)
case signal from mobile node:
case signal from ground node:
    if currentNode is sink then
        employ diversity combining with the signal copies
        received and, afterwards, decode
    else
        forward the received signal to sink
    end if
end switch
```

Algorithm 2 `Cluster`

```
Require: event ∈ {signal from mobile node, signal from ground node, cBackoffTime expired, slaveBackoff expired}
Ensure: received signal is forwarded to sink
switch (event)
case signal from mobile node:
    if received SNR > $SNR_{max}$ then
        cBackoffTime ← 0.0
    else
        map received SNR on the scale of $SNR_{diff}$ (calculated from $SNR_{max}$ - $SNR_{min}$)
        cBackoffTime ← wait time from the scale
    end if
    start cBackoffTime
case cBackoffTime expired:
    broadcast CH selection and start slaveBackoff
case signal from ground node:
    if currentNode is sink then
        employ diversity combining with the signal copies received and, afterwards, decode
    else if CH selection broadcast then
        forward received signal copy to CH
        cancel cBackoffTime
    else
        forward the signal to sink
    end if
case slaveBackoff expired:
    employ diversity combining with the signal copies received, decode, and forward the result to sink
end switch
```

Algorithm 3 `Tree`

```
Require: event ∈ {signal from mobile node, signal from ground node, tBackoffTime expired}
Ensure: received signal is forwarded to sink
switch (event)
case signal from mobile node:
    tBackoffTime ← level ∗ baseBackoffTime
    start tBackoffTime
case signal from ground node:
    if currentNode is sink then
        employ diversity combining with the signal copies received, if still not decoded
    else if signal already decoded then
        tBackoffTime ← 0.0
        start tBackoffTime
    else
        tBackoffTime ← baseBackoffTime
        start tBackoffTime
    end if
case tBackoffTime expired:
    if received more than one copy of the same signal then
        employ diversity combining
    end if
    forward the signal to sink
end switch
```

thresholds, *cBackoffTime* is the CH selection backoff time, and *slaveBackoff* is the waiting time at the CH for receiving samples before forwarding the combined signal to the sink.

In the tree approach (cf. Algorithm 3), all nodes in the ground network form a collection tree towards the sink. If signals from a mobile node are received, the *tBackoffTime* timer of a node is computed as the product of its *level* and *baseBackoffTime*, where *level* represents the ground node position relative to the sink in the network topology. On reception of signal from neighboring ground nodes, *tBackoffTime* of a node is equivalent to *baseBackoffTime*, or zero if the signal is already decoded. If any ground node encounters multiple copies of the detected data before its *tBackoffTime* is expired, it employs diversity combining before forwarding the received data, hence, resulting in a reduced output stream. Also, once the required number of data samples to successfully decode the signal are combined, the resultant data is forwarded to the sink without further delays.

IV. EVALUATION

A. *Application Scenario*

To evaluate the performance of the proposed algorithms, we target a wildlife application scenario in which the network architecture is similar to the considered heterogeneous sensor network. In brief, we focus on the BATS project that aims studying the foraging behavior of bats in their natural habitat [3]. We equip bats with a mobile node of 2 g which continuously records the contacts information between individuals. To extract this saved information, we deploy a ground network in the hunting area of bats. Whenever a bat visits the hunting area, the mobile node receives a periodic wake-up signal from the ground network and then starts transmitting recorded information to the nearby ground nodes. In the case of detection, the participating ground nodes forward only the selective signal samples that correspond to the received signal with a high probability to the central sink for diversity combining and final decoding [6].

We implement a two-dimensional bat mobility model as discussed in [15] in an area of $1000\,\mathrm{m} \times 1000\,\mathrm{m}$ including a hunting area of $300\,\mathrm{m} \times 300\,\mathrm{m}$ in the center. The ground network is placed in the hunting area with a total of 121 nodes in form of a grid with an inter-distance of 30 m as shown in Figure 2. We divide the ground network into three regions, i.e., near, middle, and far, based on their distances from the sink for a more detailed study. We also assume that nodes in the network are synchronized up to a level of ms using Network Time Protocol (NTP) [16]. A bat always starts its movement from top left corner of the simulated ground network based on the adopted Levy flight mobility model and transmits every 100 ms with a power of −47 dBm using a packet size of 12 B that also contains a 2 B of training data (preamble) for channel estimation [6]. The mobile node transmits with a data rate of

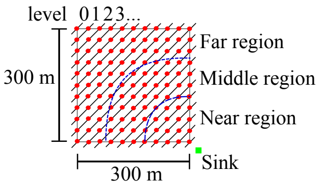


Figure 2. Overview of the simulated ground network with 121 nodes placed in a grid highlighting three considered regions.

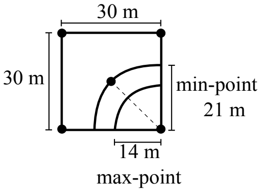


Figure 3. Schematic of a sub-grid with four ground nodes with an inter-distance of 30 m to show the max-point and min-point.

200 kbit/s according to the BATS protocol described in [3]. For communication within ground network, nodes rely on standard Wi-Fi. To simulate the wireless channel, we include FSPL based on the distance measures and linear fading loss of 0.25 dB/m.

B. Implementation and Calibration

We implemented all three algorithms, whereby the centralized algorithm is rather straight forward and passing of information from the ground nodes to the centralized sink is based on static routing.

For the cluster algorithm, the first important task is to choose a CH. Whenever a bat signal is detected by the ground network, the involved nodes use the SNR to decide if they want to become a CH. We then calculate two points around a ground node, i.e., max-point and min-point as shown in Figure 3. At any time instant, a bat can be within a max-point of only one ground node, hence, leading to a SNR at that node greater than SNR_{max} and easily declaring this node as a CH. Here, SNR_{max} (in dB) is calculated as

$$SNR_{max} = (P_{\text{tx}} - L_{\text{FadingMax}} - L_{\text{FreeSpaceMax}}) - P_{\text{noise}}, \quad (1)$$

where P shows the power levels and L represents the different loss terms. Similarly, if the SNR is less than SNR_{max} but greater than SNR_{min}, where SNR_{min} (in dB) is calculated as

$$SNR_{min} = (P_{\text{tx}} - L_{\text{FadingMin}} - L_{\text{FreeSpaceMin}}) - P_{\text{noise}}, \quad (2)$$

the respective ground node can be a contender for a possible CH because of sharing the same area with neighboring ground nodes. For the final selection of a CH among the contender nodes, we calculate $SNR_{diff} = SNR_{max} - SNR_{min}$ and then divide it into equal parts discretizing using the distance in meter between the two mentioned points. Every contender node finds *cBackoffTime* (in ms) by mapping its received SNR on the scale of SNR_{diff} as depicted in Figure 4. The contender node then broadcasts a CH announcement when its *cBackoffTime* expires and if it has not received another announcement. Once a CH is broadcast, nodes receiving the broadcast join the cluster to become CMs and forward their signal copies to the CH. The CH waits for *slaveBackoff* time to receive the responses of its CMs before it finally employs diversity combining and forwards the resulting signal as a single message towards the sink. We set the *slaveBackoff* time to 10 ms in our implementation as the ground network can receive at most once per that time interval according to the BATS protocol [3].

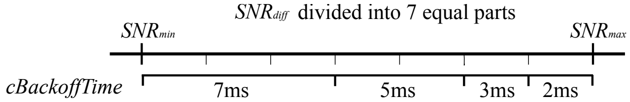
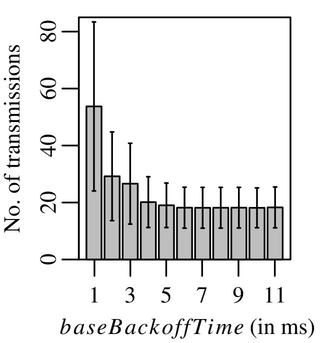


Figure 4. Mapping of received SNR on the scale of SNR_{diff} to find the *cBackoffTime*.



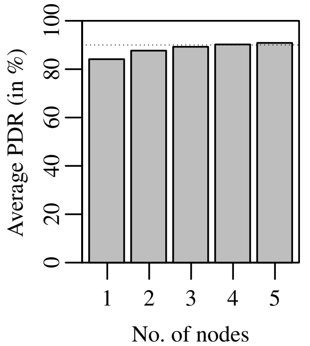


Figure 5. Average number of transmission to find optimal *baseBackoffTime* for the tree algorithm.

Figure 6. Achieved PDR with different number of ground nodes or diversity branches.

For the tree algorithm, we define a *level* in the ground network based on its topology shown as the diagonal lines in Figure 2. The order of *level* increases when we move towards the sink. The *tBackoffTime* of a node is defined as the product of its *level* and *baseBackoffTime*. Ground nodes wait for *tBackoffTime* after receiving a signal from the mobile node. Once it expires, they forward the received data towards the sink. The value of level is lower at farther nodes (integer value starting from zero) because we want nodes farthest from sink receiving the mobile node signal to forward it earlier in the network than the nodes that are nearer to the sink. With that, there is a high probability of employing diversity combining early in the network. Other ground nodes on reception of signal from neighboring ground nodes wait for only *baseBackoffTime* before further forwarding the received signal. Obviously, *baseBackoffTime* is a critical value. If too small, ground nodes will forward their received signal before copies from nodes at a lower level arrive; if too large, the end-to-end time delay will increase. We thus empirically identified a good backoff time for our scenario. In particular, we recorded the average number of transmissions within the ground network for routing the received signal copies completely to the sink. The results are shown in Figure 5. As can be seen, the number of transmissions in the network stops reducing further after 6 ms and the standard deviation stabilizes, hence, making it an optimal *baseBackoffTime* for this topology.

Before discussing the performance of these algorithms, we performed additional simulation experiments to study the possible gain of diversity combining for different numbers of received signal samples. In our previous work [6], [17], we studied the expected PDR for different SNRs both in simulation as well as in experiments. Therefore, in this work,

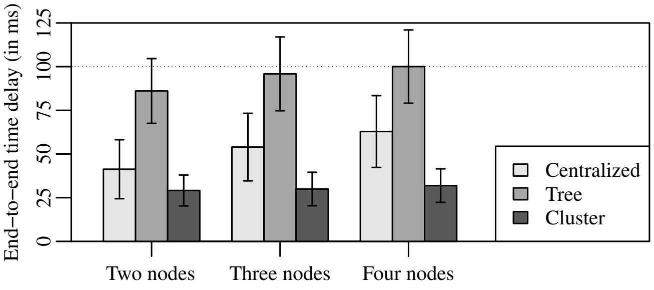


Figure 7. Average end-to-end time delay for the different algorithms to combine at least two, three, or four signal copies.

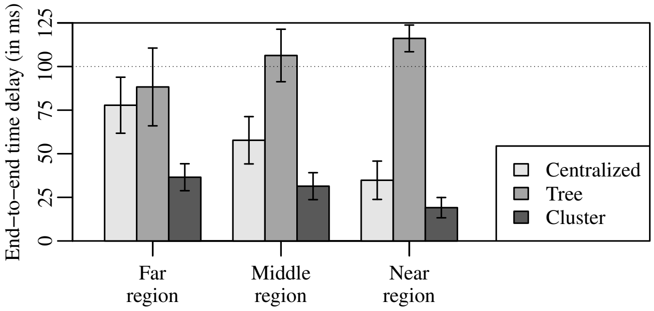


Figure 8. Average end-to-end time delay for the different algorithms to combine at least four signal copies of bat transmissions in different regions.

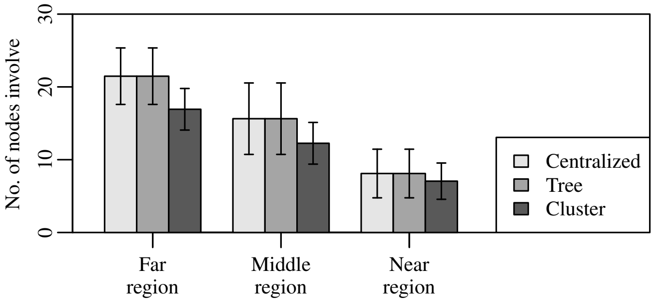


Figure 9. Total number of ground nodes involved for the different algorithms to forward the signal successfully to the sink.

we consider different bat positions in the simulated area and calculate the received SNR at different ground nodes for every transmission. We then apply Maximum Ratio Combining (MRC) (i.e., the weighted sum of all signal copies) and plot the resulting PDR. Figure 6 shows the PDR achieved with different number of nodes involved. As an example, we highlight a PDR of 90 % by a horizontal dotted line. With five ground nodes (i.e., five diversity branches), the achieved PDR reaches 91 % in comparison to a single node that achieves a PDR of only about 84 % without any diversity combining. In the following, we focus on the overall application layer performance of the different collection algorithms.

C. Results and Discussion

To compare different considered algorithms, first, we analyze the end-to-end time delay, i.e., the time it takes from transmission by the mobile node to reception at the sink. Figure 7 shows the mean end-to-end time delay of the considered algorithms when combining at least two, three, or four signal copies. The error bars show the standard deviation; for better interpretation, the horizontal line highlights a delay of 100 ms.

For the centralized algorithm, combining more signal copies leads to increased time delays due to the additional waiting time required for more transmissions to reach to the sink. It is interesting to see that the mean end-to-end delay for the cluster algorithm always stays around 30 ms. The overhead of collecting more signal copies is marginal because, once a cluster is formed, it takes a negligible amount of time to get data from an additional ground node. The tree algorithm, on average, takes more time compared to other algorithms as it involves overhead due to the backoff at every ground node.

Since the plot shows average time delays for bat transmissions over the whole network area, we further study the protocol behavior in the mentioned regions (cf. Figure 2) in Figure 8. We can see that the more a bat transmission happens near the sink, the less is the end-to-end time delay for the cluster and the centralized algorithm. However, for the tree algorithm, an opposite effect can be seen. This is due to the *level*-based backoff calculation. A transmission far from the sink provides a high chance for early signal combination and, hence, forwarding the resultant signal with a small delay. However, in the case of transmission near to the sink, every node has to wait for a longer backoff due to the higher level order involved, resulting in longer end-to-end delays.

Secondly, we investigate the number of nodes involved as a metric for the energy load distribution. Figure 9 shows the total number of ground nodes that participate either in processing or forwarding the received bat signal for all algorithms. The error bars depict the standard deviation of the mean values. The tree and centralized algorithms show a similar performance and involve between 8 and 21 ground nodes, depending on the region. The cluster algorithm performs slightly better and uses only between 5 and 18 nodes.

Next, we compare the number of transmissions in the ground network for each considered algorithm. The results are shown in Figure 10. Even though the tree algorithm involves the same number of nodes as the centralized algorithm and end-to-end time delay is much higher than the other two algorithms, the total number of transmissions is very low. The performance of tree is marginally better than the cluster algorithm in near region because cluster involves additional transmissions for forming the cluster. These additional transmission are negligible in comparison to the higher number of network transmissions in other regions. The tree and cluster algorithms involve about three times less transmissions compared to the naïve centralized one.

Finally, we consider the channel utilization of ground nodes in all three regions for bat transmissions over the whole network area and plot an eCDF in Figure 11. It can be seen that the channel utilization of the cluster algorithm is always better or equivalent to the tree algorithm but much better compared

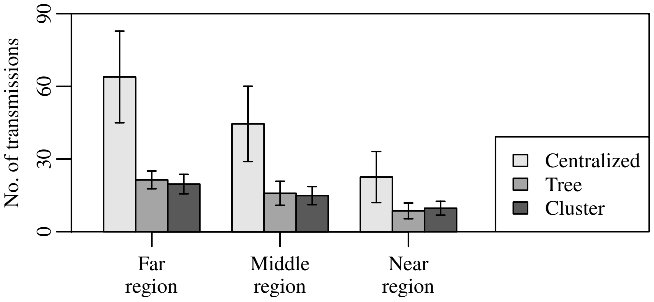


Figure 10. Total number of transmissions in the ground network for the different algorithms to forward the signal successfully to the sink.

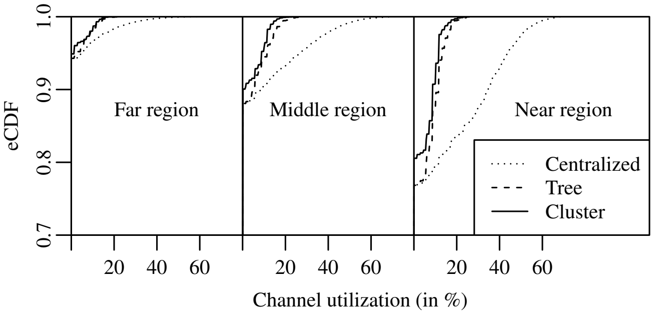


Figure 11. Channel utilization of ground nodes in different regions.

to the centralized algorithm in all cases. This can be well explained by the fact that once a cluster combines all signal copies, the resulting signal is usually routed on a single path towards the sink. However, for the centralized algorithm, many of the nodes are involved in forwarding multiple signal copies through different routes in the network resulting in the worst channel utilization.

V. CONCLUSION

In this work, we proposed two algorithms for performing distributed diversity combining in a sensor network. In particular, we developed both a cluster and a tree based solution to collect received signal samples and, instead of sending those to a central sink node, applying diversity combining early in the collection process. The main objective was to reduce the load in the sensor network. Using a wildlife monitoring application as a realistic example, we calibrated the protocols accordingly to study the performance gains in an extensive set of simulations. Our results clearly show that even though the tree algorithm is energy-efficient and reduces the network load, its performance is the worst when looking at the end-to-end delay. In contrast, the cluster algorithm outperforms all other considered algorithms not only in terms of energy efficiency due to reduced number of transmissions in the network but also provides better channel utilization.

ACKNOWLEDGEMENTS

This work has been supported by the German Research Foundation (DFG), grant no. FOR 1508.

REFERENCES

[1] C. Rutz, Z. T. Burns, R. James, S. M. Ismar, J. Burt, B. Otis, J. Bowen, and J. J. S. Clair, "Automated mapping of social networks in wild birds," *Current Biology*, vol. 22, no. 17, pp. R669–R671, 2012.

[2] Z. He, R. Kays, Z. Zhang, G. Ning, C. Huang, T. X. Han, J. Millspaugh, T. Forrester, and W. McShea, "Visual Informatics Tools for Supporting Large-Scale Collaborative Wildlife Monitoring with Citizen Scientists," *IEEE Circuits and Systems Magazine*, vol. 16, no. 1, pp. 73–86, Firstquarter 2016.

[3] F. Dressler, B. Bloessl, M. Hierold, C.-Y. Hsieh, T. Nowak, R. Weigel, and A. Koelpin, "Protocol Design for Ultra-Low Power Wake-Up Systems for Tracking Bats in the Wild," in *IEEE International Conference on Communications (ICC 2015)*. London, UK: IEEE, Jun. 2015, pp. 6345–6350.

[4] D. Gesbert, S. Hanly, H. Huang, S. S. Shitz, O. Simeone, and W. Yu, "Multi-Cell MIMO Cooperative Networks: A New Look at Interference," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 9, pp. 1380–1408, Dec. 2010.

[5] F. Kirsten, D. Öhmann, M. Simsek, and G. P. Fettweis, "On the utility of macro- and microdiversity for achieving high availability in wireless networks," in *26th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2015)*. Hong Kong, China: IEEE, Aug. 2015, pp. 1723–1728.

[6] M. Nabeel, B. Bloessl, and F. Dressler, "Efficient Receive Diversity in Distributed Sensor Networks using Selective Sample Forwarding," *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 2, pp. 336–345, Jun. 2018.

[7] M. M. Zanjireh and H. Larijani, "A Survey on Centralised and Distributed Clustering Routing Algorithms for WSNs," in *81rd IEEE Vehicular Technology Conference (VTC2015-Spring)*. Glasgow, UK: IEEE, May 2015.

[8] J. Leu, T. Chiang, M. Yu, and K. Su, "Energy Efficient Clustering Scheme for Prolonging the Lifetime of Wireless Sensor Network With Isolated Nodes," *IEEE Communications Letters*, vol. 19, no. 2, pp. 259–262, Feb. 2015.

[9] P. Nayak and A. Devulapalli, "A Fuzzy Logic-Based Clustering Algorithm for WSN to Extend the Network Lifetime," *IEEE Sensors Journal*, vol. 16, no. 1, pp. 137–144, Jan. 2016.

[10] T. Kuo and M. Tsai, "On the construction of data aggregation tree with minimum energy cost in wireless sensor networks: NP-completeness and approximation algorithms," in *31st IEEE Conference on Computer Communications (INFOCOM 2012)*. Orlando, FL: IEEE, Mar. 2012, pp. 2591–2595.

[11] H. Lin and W. Chen, "An Approximation Algorithm for the Maximum-Lifetime Data Aggregation Tree Problem in Wireless Sensor Networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3787–3798, Jun. 2017.

[12] R. Velmani and B. Kaarthick, "An Efficient Cluster-Tree Based Data Collection Scheme for Large Mobile Wireless Sensor Networks," *IEEE Sensors Journal*, vol. 15, no. 4, pp. 2377–2390, Apr. 2015.

[13] W.-S. Jung, K.-W. Lim, Y.-B. Ko, and S.-J. Park, "A Hybrid Approach for Clustering-Based Data Aggregation in Wireless Sensor Networks," in *3rd International Conference on Digital Society (ICDS 2009)*. Cancun, Mexico: IEEE, Feb. 2009, pp. 112–117.

[14] K. Shim and C. Park, "A Secure Data Aggregation Scheme Based on Appropriate Cryptographic Primitives in Heterogeneous Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 8, pp. 2128–2139, Aug. 2015.

[15] F. Dressler, M. Mutschlechner, B. Li, R. Kapitza, S. Ripperger, C. Eibel, B. Herzog, T. Hönig, and W. Schröder-Preikschat, "Monitoring Bats in the Wild: On Using Erasure Codes for Energy-Efficient Wireless Sensor Networks," *ACM Transactions on Sensor Networks*, vol. 12, no. 1, Feb. 2016.

[16] D. L. Mills, "Internet Time Synchronization: the Network Time Protocol," *IEEE/ACM Transactions on Networking (TON)*, vol. 39, no. 10, pp. 1482–1493, Oct. 1991.

[17] M. Nabeel, B. Bloessl, and F. Dressler, "On Using BOC Modulation in Ultra-Low Power Sensor Networks for Wildlife Tracking," in *IEEE Wireless Communications and Networking Conference (WCNC 2016)*. Doha, Qatar: IEEE, Apr. 2016, pp. 848–853.