Asymmetric Multi-way Ranging for Resource-Limited Nodes

Erik H.A. Duisterwinkel[1,2](✉), Niels A.H. Puts[2], and Heinrich J. Wörtche[1,2]

[1] INCAS3, Assen, The Netherlands
erikduisterwinkel@incas3.eu
[2] Eindhoven University of Technology, Eindhoven, The Netherlands

Abstract. Cooperative localization in WSN is used in applications where individual nodes cannot determine their location based on external contact, like e.g. GPS. The applications we focus on are the exploration and mapping of flooded cavities that are otherwise inaccessible or difficult-to-access, e.g. underground (oil-) reservoirs or industrial tanks for e.g. mixing. High levels of miniaturization are required for the nodes to traverse these cavities; nodes will have to be stripped down to a bare minimum. Ultrasound time-of-flight is used as radio communication is infeasible. Network topology is highly unpredictable and fast changing.

We present an asymmetric multi-way ranging protocol for these highly resource-limited, miniaturized, autonomous nodes. The specific set of constraints imposed by these applications, like the use of ultrasound, high latency, low data-rates, and non-static network topology is far-reaching and has not been studied before. Simulations of the protocol show trade-off's that can be made between ranging latency, signal overlap and overall energy budget.

Keywords: WSN · Sensor swarm · Multi-way ranging · MWR · Resource-limited

1 Introduction

Underground cavities like (oil-)reservoirs, mines and geothermal sources, and industrial infrastructure like, pipelines, mixing tanks and reactors are systems which have in common that they are hard to access for in situ measurements of system structure, dynamics, conditions and integrity. A straight forward approach which has been proposed and investigated in [1,2] is based on directly injecting large quantities of miniaturized sensor systems ('sensor motes') into the flooded system[1], let them go with the flow in order to penetrate and to explore the system as visualized in Fig. 1.

For these sensor nodes to pass through the environment and explore it without disturbing it or interfering with the dynamics, the nodes need to be scaled

[1] In this paper, *mote* and *node* will be used interchangeably, as well as *system* and *environment*.

Y. Zhou and T. Kunz (Eds.): ADHOCNETS 2016, LNICST 184, pp. 50–63, 2017.
DOI: 10.1007/978-3-319-51204-4_5

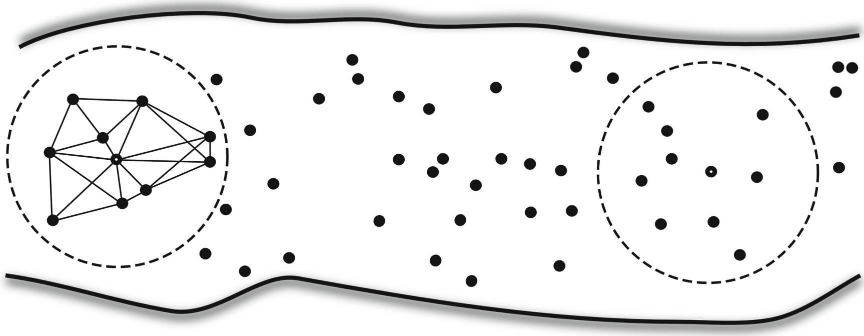


Fig. 1. Swarm of exploring sensor nodes forming a network within an enclosed environment. Nodes perform ranging transactions to neighbouring nodes within their communication range for localization and further analysis offline.

down to the centimeter or millimeter scale, depending on the application. This highly limits the resources, like energy, processing and memory, that can be taken on board the nodes. Furthermore, antennas with those dimensions will only efficiently produce radio signals with wavelengths that have an extreme high attenuation in the liquids in these environments, effectively blocking all radio communication. However, ultrasound transducers at these scales do provide larger communication ranges in these environments, but yields other problems for stable and fast communication between dynamic nodes in enclosed environments [3]. Instead of relying on communication of data, measured data is stored in memory and made available for offline analysis after retrieving the nodes from the environment.

A crucial requirement is to obtain knowledge of the positions of the nodes while traversing the environment. Structural information can be extracted from this and sensor measurement of relevant parameters (e.g. temperature, pressure, salinity) can be visualized on a map. However, during operations, neither a distributed system of anchor points nor external beacons will be available. The concept of cooperative localization [4] can be used; nodes perform measurements like time-of-flight (TOF), angle-or-arrival (AOA) or received signal strength (RSS), to gain knowledge about the position of nodes relative to neighbouring nodes within communication range. This paper introduces a *ranging protocol* to determine distances between nodes using round-trip TOF that can be used for *localization algorithms* like in [5–7], but under the specific constraints that are found in these applications.

Besides the limitation on the nodes resources, localization is further hindered by the fact that network topology – the nodes' positions and their (sparse) connectivity to neighbouring nodes – is non-static and highly unpredictable. As it is not known where neighbouring nodes are positioned, omni-directional ultrasound is used for ranging measurements. This can be achieved using e.g. tube-shaped transducers as in [8].

Ultrasound is often used for ranging applications as the propagation speed is 10^5 times lower than that of radio, therefore allowing for larger timing errors. However, the low propagation speed in combination with the significantly lower data transmission rates (typically 2–40 kb/s or even lower in more challenging environments), introduces challenges that are less often seen in radio communication [3]. Latency in the ranging transactions makes that the movement of the nodes becomes significant in the distance determination. The low data-rates, in combination with the enclosed environment and non-static topology makes signal overlap a significant hinder.

In this paper we present a novel asymmetric multi-way ranging protocol, in which trade-offs are made between the energy budget, the ranging latency and the signal overlap to optimally use the on board resources for obtaining nodes positions in offline analysis for the above mentioned applications. Depending on the application or the state that nodes are in, these trade-off's can be adjusted to address the specific situation as good as possible.

The specific challenges in developing the ranging protocol for these applications are addressed in Sect. 2. In Sect. 3 the design of a ranging protocol is described that attempts to balance between all the parameters involved. In order to assess the suitability of the protocol for these applications, the protocol is simulated in a network simulator as described in Sect. 4. Important performance metrics that assess the specific goals are shown in the results Sect. 5. Discussion and future work can be found in Sect. 6, the conclusion in Sect. 7.

2 Protocol Design Challenges

Traditional ranging protocols consist of three phases: a scanning phase, a ranging phase and a reporting phase [9]. In this paper, an attempt is being made to maximally reduce energy costs of ranging in the specific application cases described above.

Reporting of ranging measurements to neighbouring nodes is not performed as this would require extra node resources and data communication is challenging in this applications. Nodes only store measurements in their own memory.

The ranging-phase can be performed in a variety of methods. We chose for the concept of multi-way ranging (MWR), initially proposed in [10] as N-Way Time Transfer. It exploits the omnidirectional transmission and reception by using all received signals for determining distances between nodes, rather than only the signal between sender and one addressed receiver in e.g. two-way ranging (TWR) methods [11]. Therefore, the total amount of messages needed to complete a full ranging cycle using MWR scales linearly with the number of nodes, instead of quadratically in TWR methods. It significantly reduces the energy required for performing the ranging procedure.

Control of Ranging Sequence. As the network topology is non-static and connectivity sparse and fast-changing, it is not known which neighbouring nodes are within communication range. The simple sequence of events in traditional

MWR [10], where node $i+1$ transmits a ranging signal after node i, cannot be easily controlled in these applications. An alternative method is proposed in Sect. 3 using a master-slave system.

Ranging Latency. The ranging latency of a single ranging transaction takes up to 3 ms when nodes are 1 m apart (twice the propagation time, the message length and the processing time). A full ranging cycle within a large swarm, with all its individual ranging transactions, can easily take 100 ms. Depending on the movement of the nodes, a large latency significantly challenges the localization algorithm as the measured inter-node distances cannot be considered quasi-static. The latency should therefore be kept as low as possible.

Furthermore, from an energy perspective it is beneficial to reduce latency such that nodes are longer in a low-energy sleep state instead of an active listening/decoding state.

Signal Overlap. The low data-rates in combination with small inter-node distances in enclosed environments cause a significant amount of potential signal overlap. Signal overlap should be prevented as much as possible as it requires more energy and processing to filter and distinguish signals. The ranging protocol in Sect. 3 uses a time-divided communication scheme for determination of the distances to allow for reduction of signal overlap. The amount of bits transmitted should also be kept at a minimum to keep the message length as short as possible to reduce signal overlap.

Scanning Phase. As it is not known which nodes are within communication range, often in ranging protocols, a separate scanning phase is initiated before the ranging phase. In this phase, nodes determine which neighbouring nodes are within communication range to determine which nodes to perform ranging measurements to.

Such an additional scanning phase adds to the energy budget. In this paper, the scanning phase is omitted and solved by addressing all nodes by a non-unique calling identifier. It causes a trade-off between ranging latency and signal overlap.

3 Protocol

This section introduces a modified version of the regular multi-way ranging protocol to deal with the specific limitations in the usage of ultrasound in a non-static network topology with highly resource-limited nodes. It will also address the challenge of finding proper trade-off's between e.g. ranging latency and signal overlap. It is important to notice that these trade-off's can be adjusted based on the specific environments or the specific situation that nodes are in.

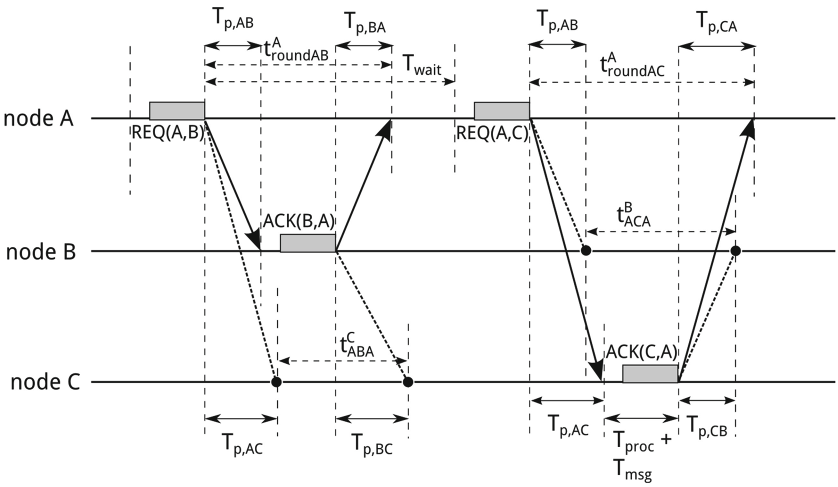


Fig. 2. One ranging cycle of master node A and slave nodes B and C. The master node transmits request (REQ) signals and slave nodes respond with and acknowledgement (ACK) signal if it is addressed to them. Knowledge of timing information $t^{A}_{roundAB}$, $t^{A}_{roundAC}$, t^{C}_{ABA} and t^{B}_{ACA} and the fixed value of the processing time T_{proc} is sufficient to determine the propagation times $T_{p,AB}$, $T_{p,AC}$ and $T_{p,BC}$ between the nodes.

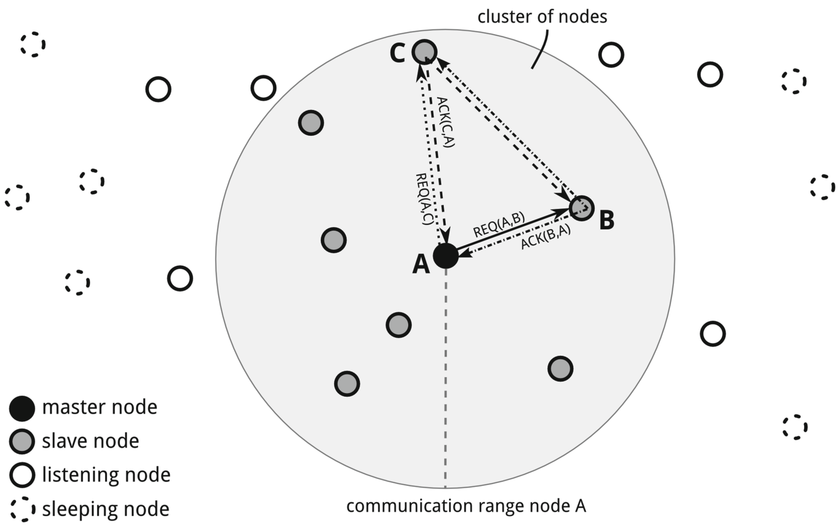


Fig. 3. Master node A initiates the ranging process to node B and C. The nodes within communication range become slave node and respond to REQ signals with an ACK signal. All nodes within the respective communication range receive the signals and store them: in offline analysis they can be used to determine round-trip TOF between nodes. The nodes outside the cluster will only receive ACK signals. In this figure, not all signals (arrows) are drawn.

3.1 Asymmetric Multi-way Ranging

Instead of traditional, 'symmetric', MWR as introduced in [10], here the ranging procedure is controlled by *master nodes* that send request (REQ) signals to its neighbouring nodes that then become *slave nodes* and respond with an acknowledgement (ACK) signal. The communication scheme that is used is illustrated in Fig. 2. The *cluster of nodes* that is formed by this master node and the slave nodes is illustrated in Fig. 3.

In a ranging transaction between master node A and slave node B, the timestamps of transmission and reception at node A provides knowledge about the round-trip TOF between nodes A and B, denoted as $t^A_{roundAB}$. The node's internal processing time T_{proc} and signal message time T_{msg} are known beforehand and are fixed, therefore, the round-trip propagation time, $T_{p,AB} + T_{p,BA}$, between A and B can be estimated. After this ranging transactions, master node A performs a similar transaction to node C and the other nodes within the cluster.

Since the nodes A, B and C are within each others communication range, also the nodes that are not addressed in the ranging transactions receive the signals. The time difference t^C_{ABA} between the arrival of REQ(A,B) and ACK(B,A) at node C and the time difference t^B_{ACA} between the arrival of REQ(A,C) and ACK(C,A) at node B, can be used to calculate the propagation time between nodes B and C using:

$$T_{p,BC} + T_{p,CB} = t^C_{ABA} + t^B_{ACA} - 2(T_{\text{proc}} + T_{\text{msg}}) \tag{1}$$

3.2 Picking the Master Node

The role of master node is being alternated between all nodes in the network. The advantage of this is that the power consumption is distributed evenly over all nodes (master nodes transmit more signals) and clusters are more distributed over the swarm.

Within the time frame of one sample T_{sample}, in which a complete ranging cycle is completed for all nodes, the role of master node is chosen randomly. This is performed by having all nodes at the beginning of a sample chose a random delay time T_D. Nodes become master when their sample timer t_s, that is set to zero at the beginning of a sample, trespasses $t_s > T_D$. Nodes become slave node when before t_s reaches T_D, it receives any REQ signal from a master node. The master node initiates the ranging transactions as described above, thereby forming a cluster of nodes as in Fig. 3.

As seen in Fig. 1, throughout the entire network, several of these clusters are formed in which ranging transactions are performed. Every sample, these clusters change based on which nodes have become master node.

3.3 Scanning the Slave Nodes

Within one ranging cycle, the master node should send a request to all slave nodes in the cluster, but it is beforehand not known which nodes are within communication range. Regular scanning techniques depend on the availability of sufficient bandwidth, processing power or time to perform broadcasting.

In this work we propose for the master node to initiate the ranging transactions to all possible hardware addresses. But as the total amount of nodes in the network can be very large and the connections are sparse, this will be very inefficient as most requests remain unanswered. Instead of requesting to the hardware's *unique identifiers* (UID) the master node requests to highly abbreviated *calling identifiers* (CID). The master node only initiates n_f times a ranging transaction to CID $= \{0, 1, ..., n_f\text{-}1\}$. Slave nodes will respond if and only if their unique hardware identifier suffices

$$\mathrm{mod}(\text{UID}, n_f) = \text{CID} \tag{2}$$

As multiple nodes will have an identical CID, the probability arises that multiple nodes will respond to the same request. If the ACKs of the responding slave nodes do not overlap such that the signals cannot be distinguished and decoded anymore at the receiving node, the determination of the round-trip TOF of each of them can still be performed. Parameter n_f can be chosen both offline as online to adjust for the amount of neighbouring nodes and the total signal overlap.

In order to receive all possible ACK's, the master node will wait $T_{\text{wait}} = 2T_{\text{p,max}} + T_{\text{proc}} + T_{\text{msg}}$ after transmission of a REQ before it sends a request with a next CID. Here, $T_{\text{p,max}}$ accounts for the propagation time required to reach the end of the (expected) communication range.

After all ranging transactions have been performed, the nodes will go into a low-energy sleep mode to await the start of the next sample. When $t_s > T_{\text{sample}}$, nodes will internally initiate a new sample. The sample is initiated in a sleep mode and nodes will wake up upon reception of any signal (using e.g. a threshold detection). It then starts a listening mode in which it can decode incoming signals. Before a master node starts with the first CID, it can transmit a short signal to wake up the neighbouring nodes.

3.4 Reducing the Latency

As seen in Fig. 3, the nodes just outside the communication range of the master node do not become slave node and will have to wait for itself to become master node, or will have to wait for a node within its communication range to become one.

In order to speed up this process and have the network-wide ranging cycle end sooner, an avalanche effect is induced. Nodes that receive ACK signals without having received REQ signals are likely to be just outside an already formed cluster. Their remaining delay time before they become master node is reduced by a factor $M_{\text{avalanche}}$ at the reception of any ACK signal until they become master or slave.

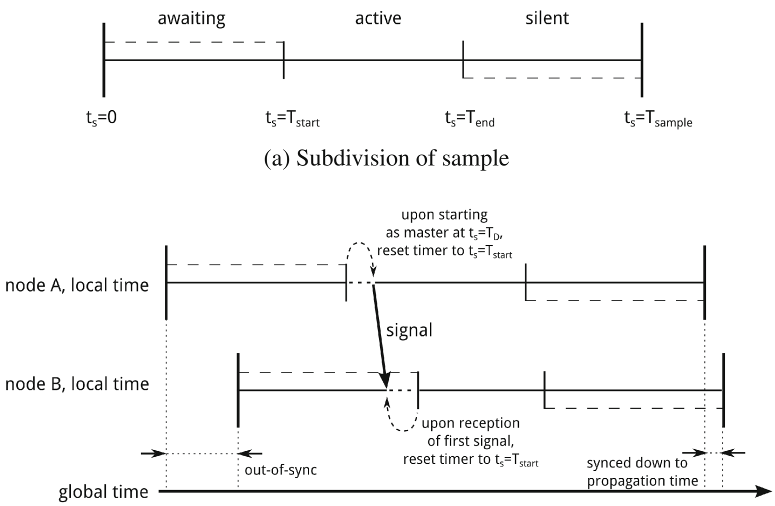
This reduction of the delay time T_D induces an avalanche effect throughout the network such that all nodes become either master or slave within less time after each other, therewith, reducing the ranging latency throughout the network.

3.5 Synchronization

Absolute synchronization is not required for determining distances as all distances are obtained using a round-trip TOF measurement. It is however beneficial to have nodes synchronized to a level in which samples are aligned such that the avalanche effect introduced in Sect. 3.4 allows nodes to sleep for the majority of the sample time instead of responding to nodes that are still in the previous or already in the next sample.

For this reason, in order for connected nodes to remain in the same sample, it is proposed to subdivide a sample on the node level into timeslots as illustrated in Fig. 4a. The random delay time is chosen from a uniform distribution within the range $T_D \in (T_{\text{start}}, T_{\text{end}})$ or the *active time period*. The internal sample timer t_s is reset to $t_s = T_{\text{start}}$ when becoming master; or, at reception of the first signal (any REQ or ACK) in the sample, as illustrated in Fig. 4b. This will assure that connected nodes remain synchronized to the sample level, as long as (groups of) nodes have not been disconnected from each other. In the *awaiting period* nodes do not become master and can only receive signals, in the *silent period*, nodes have already received their first signal or already became master node.

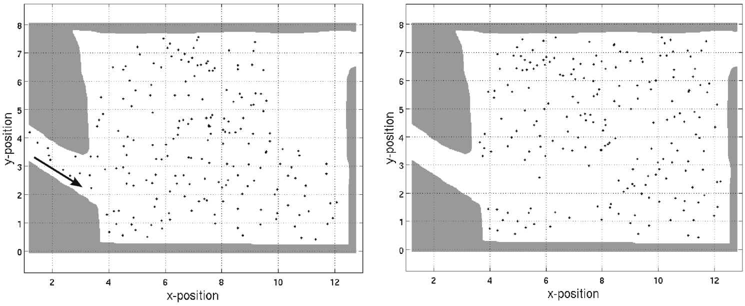
In our work, the three time periods are chosen to be of equal length, i.e. $T_{\text{start}} = \frac{1}{3}T_{\text{sample}}$ and $T_{\text{end}} = \frac{2}{3}T_{\text{sample}}$. Note that these periods do not indicate when a node is asleep or in which mode it is in.



(a) Subdivision of sample

(b) Sample timer reset, either at first reception of signal or at start as master

Fig. 4. Subdivision of sample in three parts: awaiting, active and silent. Each sample, the start delay that determines when to become master is randomly chosen within active period (uniform distribution). Upon first reception of signal in the sample, or, upon becoming master node, the sample timer t_s is reset to $t_s = T_{\text{start}}$.



(a) t=0 sec (b) t=16 sec

Fig. 5. Nodes positions throughout simulation in 2D tank-like environment with injection flow from left (indicated by arrow).

4 Simulations

The protocol implementation is simulated in OMNeT++ network simulator [12,13].

In order to simulate a dynamic swarm of nodes that passively flow through an enclosed environment, we use a flow simulator to generate the nodes positions over time [14]. The positions are generated based on tracer positions in a fluid flow in a 9 by 8 m 2-D tank-like environment with an inlet an outlet. The positions of the $N = 200$ nodes at the beginning and end of the simulation time are illustrated in Fig. 5.

The average node speed throughout the simulation is 0.20 ± 0.17 m/s with a maximum of 0.80 m/s. The communication range is set to a fixed 1 m and results in an average node density of 9.8 ± 3.7 neighbouring nodes within the communication range. The clock frequency offset is set to 100 ppm and is fixed throughout the simulation.

The sample time is set to $T_{\text{sample}} = 1$ s and the amount of CIDs is in this paper is swept between: $n_f = \{8, 16, 32\}$. The ultrasound transmission rate is set to 40 kb/s. The avalanche induction is studied by sweeping $M_{\text{avalanche}} = \{1, 2, 4, 8\}$ in which $M_{\text{avalanche}} = 1$ means no induced avalanche.

The output of the simulations consist of the data that are being stored on the nodes internal storage: the messages sent and received, the timestamp at transmission/reception, the timestamp when new samples start and for research purposes also the internal states the nodes are in and at what specific time.

5 Results

The protocol is analysed based on several performance metrics that assess the design goals for the ranging method. One is the ranging latency within a swarm versus the signal overlap, second is the energy usage of an individual node. And

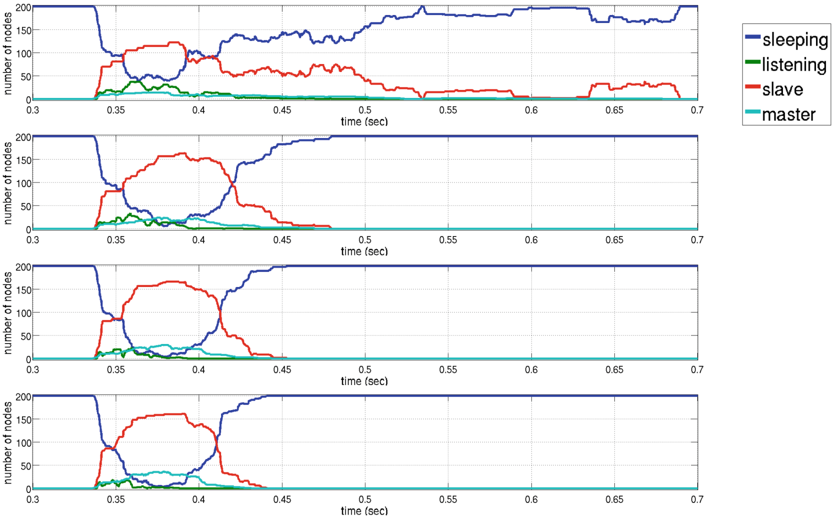


Fig. 6. Total number of nodes in specific states through first sample. From top to bottom the avalanche parameter: $M_{\text{avalanche}} = 1, 2, 4, 8$. The top graph, $M_{\text{avalanche}} = 1$ means there is no induced avalanche effect. In these simulations, the number of CIDs $n_f = 16$.

as last, the fraction of the theoretical amount of possible distances that are determined using this protocol: the coverage.

Figure 6 shows for the simulated datasets with $n_f = 16$ an overview of the amount of nodes in a specific state over a single sample. Figure 7 shows the main performance metrics of the protocol in a simulated network as described in Sect. 4 and discussed next.

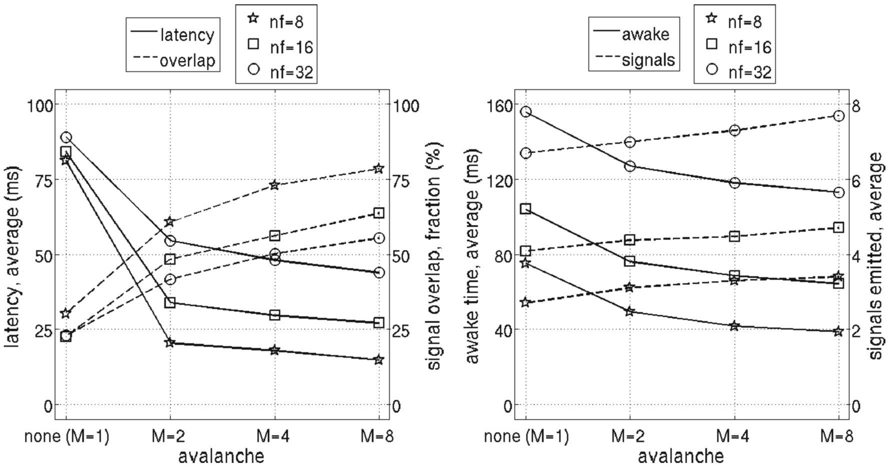
5.1 Latency Versus Signal Overlap

From Fig. 6 it can be seen that the induced avalanche effect assures that the network finished a single ranging cycle sooner. In this sample, the maximum latency goes from 360 ms for $M_{\text{avalanche}} = 1$ (no avalanche) down to 150 ms, 120 ms and 110 ms for $M_{\text{avalanche}} = \{2, 4, 8\}$, respectively. As a reference; within a single cluster, the ranging latency is $n_f T_{\text{wait}} = 43$ ms.

Figure 7a shows the average latency between all ranging transactions in a ranging cycle. Increasing the avalanche effect ($M_{\text{avalanche}}$) yields a smaller latency. The latency of the full ranging cycle is approximately between 4 to 5 times larger as the average latency between the transactions.

Figure 7a also shows the average fraction of signals that are received with overlap with another signal. Increasing the avalanche effect and reducing the latency inevitably increases the signal overlap.

At lower values of n_f, the latency drops quicker, but signal overlap is higher; less CIDs are scanned but more nodes will respond to the same REQ signal. There is a clear trade-off between latency and signal overlap.



(a) Ranging latency and signal overlap (b) Awake time and signals transmitted

Fig. 7. Average values of perfomance metrics of simulated ranging protocol using different input parameters n_f and $M_{\text{avalanche}}$. Clear trade-offs are visible.

5.2 Energy Efficiency

The energy efficiency in this paper mainly focusses on the nodes' awake time and the amount of signals transmitted (and related to that the amount of signals received and stored). The awake time is defined as the time not spend in the low-energy sleep state, but rather in an active signal transmission or receiving/decoding state.

Figure 7b shows the node's average awake time per sample and the average amount of signals transmitted per node per sample (master and slave nodes together). As the latency decreases with increasing $M_{\text{avalanche}}$, so does the time that nodes need to be awake. With increasing avalanche effect, the number of signals required for transmission increases slightly as more nodes will become master node.

Both the awake time and the number of signals transmitted increase with increasing n_f as more CIDs will have to be transmitted and decoded.

5.3 Coverage

The coverage can be defined as the fraction of connections (node pairs that are within each others communication range) for which the ranging procedure yields sufficient information to determine a distance measure. Since only distances can be calculated within a cluster, the coverage will be lower than 100% as not all connections can fall within a cluster. Throughout all simulations, the coverage was between 86%–89%.

Even though for the other 11%–14% no distances can be determined using RT-TOF, the basic connectivity information is available: the received ACK signals that did get received by the nodes outside the cluster, provides information on which nodes where within their communication range. The localization algorithm can use this information to its advantage.

The coverage does not need to be 100% to localize the entire swarm. In fact, for example in [6], studies are performed where localization is stress-tested on e.g. the loss of large amounts of connections. Also note that each sample, different clusters are formed such that this group of 11%–14% of the connections is different for each sample.

6 Discussion and Future Work

Although the current implementation of the ranging protocol has been simulated over a relative short measurement time. Simulations using extremely large clock deviations of up to 100 000 ppm have been tested and show good alignment of samples over the simulation time of 16 s (not shown here). As long as the network is sufficiently connected and not disjoint, the avalanche effect can keep the nodes' internal clock synchronized within a fraction of the sample time T_{sample}.

The simulations in this paper have been performed in a 2-D environment. Although this protocol can be directly used in 3-D, the induced avalanche effect will have quantitatively a slightly different result as the ones presented here. No qualitative differences are to be expected.

Instead of scanning all possible CIDs, all nodes can actively record which nodes it has seen in the past. Upon becoming master node, instead of scanning all available CIDs, the master node can scan the UIDs of nodes that it has seen in the previous (several) sample periods. This will reduce signal overlap and can reduce the amount of required signals for transmission.

The protocol can fairly easily be adjusted to also account for disjoint networks coming together such that they can become synchronized up to the sample level. This is part of future work.

7 Conclusion

This paper illustrates the challenges involved in performing round-trip TOF in a swarm of autonomous nodes without external contact in an unpredictable and dynamic topology with sparse connectivity. The applications require high levels of miniaturization of the nodes and introduce a specific set of constraints and challenges that has not been researched before. A novel asynchronous multi-way ranging protocol has been presented to allow round-trip TOF measurements. Control of the ranging transactions can be performed by master nodes that initiate them to their neighbouring slave nodes. Master nodes are assigned at random each time a new sample starts.

The latency between ranging measurements in the entire swarm can be reduced by inducing an avalanche effect of nodes that become master node. The avalanche effect also reduces the required time for the nodes to be actively listening for signals and allows for synchronization down to a fraction of the sample time.

The trade-off's that are involved in this ranging protocol are a direct consequence of the application: the need for resource-limited nodes, the use of ultrasound and the unpredictable and fast-changing network topology with sparse connectivity. Getting insight in these trade-off's allow for adjusting the ranging protocol based on the specific circumstances that nodes are in. In [2], this exploration method and the ability of nodes to adjust for specific circumstances, is further explored.

Acknowledgement. INCAS^3 is co-funded by the Province of Drenthe, the Municipality of Assen, the European Fund for Regional Development and the Ministry of Economic Affairs, Peaks in the Delta. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 665347 The authors would like to thank Elena Talnishnikh, Hao Gao, Jan Bergmans, Libertario Demi and Gijs Dubbelman for their help in this research.

References

1. Talnishnikh, E., et al.: Micro Motes: a highly penetrating probe for inaccessible environments. In: Leung, H., Mukhopadhyay, S.C. (eds.) Intelligent Environmental Sensing. Smart Sensors, Measurement and Instrumentation, vol. 13, pp. 33–49. Springer, Cham (2015)
2. EU Horizon 2020 FET-Open project: PHOENIX. www.phoenix-project.eu
3. Akyildiz, I.F., Pompili, D., Melodia, T.: Underwater acoustic sensor networks: research challenges. Ad Hoc Netw. **3**, 257–279 (2005)
4. Patwari, N., et al.: Cooperative localization in wireless sensor networks. IEEE Sig. Process. Mag. **22**(4), 54–69 (2005)
5. Bachrach, J., Taylor, C.: Localization in sensor networks. In: Handbook of Sensor Networks: Algorithms and Architectures. John Wiley & Sons Inc., 23 September 2005
6. Duisterwinkel, E.H.A.: Ph.D. Dissertation, research in progress
7. Dubbelman, G., et al.: Robust sensor cloud localization from range measurements. In: IEEE International Conference on Intelligent Robots and Systems, Chicago, Illinois, USA (2014)
8. Li, H., Jung, K.W., Deng, Z.D.: Piezoelectric transducer design for a miniaturized injectable acoustic transmitter. Smart Mater. Struct. **24** (2015)
9. Kim, H.: Performance comparison of asynchronous ranging algorithms. In: IEEE Global Telecommunications Conference, Honolulu, Hawaii, December 2009
10. Green, M.P.: N-way time transfer ('nwtt') method for cooperative ranging, Contribution 802.15-05-0482-00-004a to the IEEE 802.15.4a Ranging Subcomittee, July 2005

11. Hach, R.: Symmetric double sided - two way ranging, Contribution 802.15-05-0334-00-004a to the IEEE 802.15.4a Ranging Subcomittee, June 2005
12. Puts, N.A.H.: Analysis and design of an ultrasound positioning system protocol for sensor swarms. MSc. Dissertation, Eindhoven University of Technology, The Netherlands (2016)
13. Varga, A., Homig, R.: An overview of the OMNeT++ simulation environment. In: SIMUTools, Marseille, France, 03–07 March 2008
14. Bosman, H.H.W.J.: For the adaptation and extension of Daniel V. Schroeder's Java based flow simulator (2013, 2015)