Interval-based Clock Synchronization Is Resilient to Mobility

Lennart Meier Philipp Blum Lothar Thiele

Computer Engineering and Networks Laboratory
Swiss Federal Institute of Technology (ETH) Zurich
CH-8092 Zurich, Switzerland
{meier,blum,thiele}@tik.ee.ethz.ch

***Abstract*— Clock synchronization is a crucial basic service in ad-hoc sensor networks. Most proposed algorithms employ hierarchical communication structures that may be expensive to maintain when nodes are mobile and network topologies are dynamic. Interval-based synchronization has been proposed as particularly suited for sensor networks, as it does not require any particular communication pattern. In this paper, we show that node mobility puts no burden on interval-based synchronization, but actually helps it. This is not the case for time-estimate-based approaches that rely on specific communication patterns. Mobility resilience therefore further strengthens the case of interval-based clock synchronization for ad-hoc sensor networks.**

I. Introduction

Clock synchronization is an important service in ad-hoc sensor networks. For example, the correct evaluation of distributed sensor data may require knowledge about the chronology of the sensor observations [1]. In addition, energy consumption can be reduced by synchronous power-on and shutdown of the communication circuits of a sender–receiver pair [2], [3].

Clock synchronization in ad-hoc sensor networks poses challenges that are substantially different from those in infrastructure-based networks. *Robustness:* There is no stable connectivity between nodes. *Energy efficiency:* Synchronization can only be achieved and maintained by communication, which is expensive in terms of energy. *Ad-hoc deployment:* The clock-synchronization service must not rely on any a-priori configuration or on infrastructure.

The conclusion is that algorithms such as NTP [4] cannot be directly applied in ad-hoc sensor networks. Custom-tailored algorithms have been proposed recently; most of them use time estimates. Since it is not a priori clear how two time estimates should be combined, these algorithms require a hierarchical structure in the network and prioritize the estimate of the higher-ranking node. In [5], [6], using guaranteed time intervals was presented as an alternative and particularly suitable approach. Since time intervals can be combined optimally and unambiguously by intersection, the interval-based approach does not require hierarchical structures.

The work presented in this paper was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

In this paper, we consider the impact of node mobility on clock synchronization. Examples of mobile nodes comprise nodes attached to animals or humans or nodes transported by wind or water from one location to another (environmental monitoring).

A. New results

In this paper, we extend the results from [5] by showing that interval-based synchronization is not only resilient to node mobility, but actually even benefits from it in the average case. We will argue that this is not the case for time-estimate-based synchronization approaches that employ hierarchical structures such as clusters [7] or trees [8], [9], [10], since mobility makes these structures prohibitively expensive to maintain. We introduce synchronization where nodes can use arbitrary communication partners and therefore no such overhead exists. Our simulations show that it is a viable and competitive alternative to tree-based approaches.

We then compare the performance of the algorithms analyzed in [5] for varying degrees of node mobility. Our simulations suggest that mobility enhances connectivity in sparse networks. Due to the special character of interval-based time information, mobility has only positive effects.

B. Related work

There has been much work on clock synchronization in infrastructure-based networks [11], [12]. Recent work has addressed the special characteristics of synchronization in ad-hoc networks [1], [13], [14], [15]. To our knowledge, the impact of mobility on synchronization has not yet been considered.

Interval-based synchronization provides each network node with guaranteed bounds on the real time. Using bounds instead of time estimates was first proposed in [16] and was further studied in [17]. In [5], [6], interval-based synchronization was presented as an approach that provides optimal and unambiguous propagation of time information under arbitrary communication patterns.

C. Overview

In Sect. II, we give the prerequisites for the rest of the paper. We then discuss the various possible communication patterns in ad-hoc networks in Sect. III, showing that random communications are not only a good compromise, but may be the only feasible choice in scenarios with mobile nodes. In Sect. IV, we examine the effect of node mobility on interval-based synchronization, which our simulations show to be beneficial.

II. Prerequisites

In this section, we introduce the system model and summarize results from related work that we will use.

A. Node and communication models

A network consists of nodes with local clocks that have an arbitrary offset to the real time. A clock's rate is variable but guaranteed to lie within the interval $[1-\hat{\rho}, 1+\hat{\rho}]$ for some constant $\hat{\rho}$.[1] We call this the bounded-drift model. The network also contains so-called anchor nodes. These nodes have access to real time, i.e. their rate is 1 and their offset is 0.

Each node can communicate point-to-point with any other node within its transmission range. Arguably, the absence of broadcast communication makes the distribution of data to multiple recipients more costly, and our results may not apply to networks with broadcast radios. We believe that point-to-point communication is the correct model for those sensor networks that use channel-oriented radios [18].

In sensor networks, communication is expensive in terms of energy, and hence infrequent communication is desirable [18]. Assuming infrequent communication, we can neglect delay uncertainties and eliminate them from our analysis, i.e. communication occurs in zero time. This is reasonable: As the frequency of communication decreases, the uncertainty due to clock drift increases, while the uncertainty due to message delays remains constant. Two particular characteristics of sensor nodes further strenghten the case for the dominance of the drift:

1) Time-stamping on sensor nodes can be done at a low level, such as in the MAC layer, leading to a small delay uncertainty. Recent algorithms reduce it to a few microseconds, e.g. by using packet streams or reference broadcasts
2) Sensor nodes typically employ inexpensive oscillators with drifts of up to 100 ppm.

A numeric example: If the delay uncertainty is 1 μs and the clock drift's absolute value is bounded by 100 ppm, then after 5 ms, the drift's contribution to the uncertainty equals that of the delay. After one hour, it is 720000 times larger. Even for "optimistic" values of 1 ms (uncertainty) and 10 ppm (drift), drift and uncertainty have equal impact after 50 s.

The *view* of a node at a given time is the sum of all the information that the node can potentially have acquired until that time.

[1]Typical values for $\hat{\rho}$ are in $[1\text{ppm}, 100\text{ppm}] = [0.000001, 0.0001]$.

B. Interval-based synchronization

Interval-based synchronization as introduced in [16] means that time information is represented using an interval $[T^l, T^u]$ defined by guaranteed lower and upper bounds T^l and T^u on the real time. We call the size $\Delta T = T^u - T^l$ of the interval the *uncertainty*. The goal of an interval-based synchronization algorithm is to maintain guaranteed lower and upper bounds T^l and T^u while keeping the uncertainty as small as possible.

C. General lower bound

In [5], it was shown that for any node N_i and real time t, no algorithm can guarantee a better uncertainty than

$$\Delta T \geq \min_{s \in \mathcal{S}} \left\{ (t - t_s) \left(\frac{2\hat{\rho}}{1-\hat{\rho}} \right) \right\} \quad , \tag{1}$$

where $\mathcal{S}$ is the set of all communications s with an anchor node that are in the view of node N_i [2] at time t, and t_s is the real time at which communication s takes place.[3] The expression in the braces represents the degradation due to clock drift until time t of the time information provided by the anchor node at time t_s.

D. Interval-based algorithms

A completely local interval-based algorithm was proposed in [16] as algorithm IM: Whenever two nodes communicate, they exchange their current time bounds and pick the best of the available bounds, i.e. they effectively intersect their time intervals. Current bounds are computed from previous bounds as follows: If T_1^l is a valid lower bound for real time t_1, and h_1 and h_2 are the node's local-clock readings at times t_1 and t_2, respectively, then $T_2^l = T_1^l + (h_2 - h_1)/(1+\hat{\rho})$ is a valid lower bound for real time t_2. This is a consequence of the fact that the local-time difference $h_2 - h_1$ is at most $(1+\hat{\rho})(t_2 - t_1)$ (by the bounded-drift assumption). Similarly, $T_2^u = T_1^u + (h_2 - h_1)/(1-\hat{\rho})$ is a valid upper bound on t_2. Algorithm IM is simple and its memory overhead is small, constant, and independent of the number of nodes in the network. It provides optimal uncertainty in the worst case [5].

The Back-Path Interval-Synchronization Algorithm (BP-ISA) [5] computes current bounds from previous bounds just as algorithm IM, but instead of storing only one pair of previous bounds, every node stores the bounds from the last communication with every distinct node in the network. In the average case, this leads to an improvement over algorithm IM. The BP-ISA was shown to never perform worse than algorithm IM, it is thus also worst-case optimal. Its memory overhead is constant over time, but increases proportionally with the number of nodes in the network.

[2]This means that either node N_i directly communicated with the anchor node or time information was transported from the anchor node to N_i in an implicit multi-hop communication.

[3]This lower bound can easily be transformed to a bound for estimate-based synchronization: Algorithms that compute an estimate T of real time cannot guarantee a smaller synchronization error $|T-t|$ than $\Delta T/2$. Since t can take any value in $[T^l, T^u]$, the estimate T minimizing the worst-case error is $T = 1/2(T^l + T^u)$, resulting in a worst-case error of $\Delta T/2$.

III. Communication Patterns

To achieve synchronization, time information has to be propagated through the network. A communication pattern determines which nodes communicate at which time. The time information can be embedded in the communications performed by the main application of the network or it can be transmitted in dedicated synchronization messages. In this section, we examine the latter case: We analyze different choices for communication patterns in a network with a single anchor node.

Concerning the choice of communication partners, there are two extreme structures or modes of operation:

1) The network can be organized in a tree topology with the anchor node at the root. Each parent node synchronizes with all its children. We refer to this as the tree-based mode.
2) Each node in the network synchronizes with a random node within its transmission range. We refer to this as the random-communications mode.

The tree-based mode is used by many recently proposed synchronization algorithms for ad-hoc networks, e.g. [8], [9], [10], [19]. Various methods of distributed tree construction were discussed in [10]. They are all expensive in terms of communication overhead, which increases when the nodes become mobile and the tree has to be rebuilt frequently. In contrast, the random-communications mode does not suffer from this kind of overhead at all, since it does not use any topology information.

In this section, we compare the tree-based and the random-communications modes of interval-based synchronization in terms of the maximal uncertainty in the network. In addition, we compare the distribution of the frequency of communication among the nodes in the network. A high frequency implies a high energy consumption and therefore a short node lifetime. To maximize network lifetime, it is desirable to achieve a small maximal uncertainty with few and equally distributed communications. In this section, we will show the following:

1) In a breadth-first tree, the maximal uncertainty is small, but the frequency of communication is distributed very unevenly among the nodes.
2) In a depth-first tree, the frequency of communication is distributed more evenly than in the breadth-first tree, but the maximal uncertainty is much larger.
3) For a fixed total number of communications in the network, the random-communications mode achieves an only slightly larger maximal uncertainty than the breadth-first tree, while communications are distributed as evenly as in the depth-first tree.
4) For a fixed maximal number of communications per node, the random-communications mode achieves a strictly smaller uncertainty than either tree-based mode.

A. Specialized lower bounds

We now show how the uncertainty in the tree-based mode can be compared to the one achieved with random communications. To this end, we derive specific lower bounds on the uncertainty from (1).

1) Lower bound in a tree: Let a tree with an anchor node at the root be given, and let f_C be the frequency of communications per node. If every parent node communicates with all its child nodes at least every $1/f_C$, then for all children of the anchor node, the real-time difference $(t - t_s)$ between any time t and the time t_s of the latest communication with the anchor node before time t is at most $1/f_C$. For an arbitrary node, the difference $t - t_s$ is at most $(\textit{hop distance to root})/f_C$. Thus, the worst-case uncertainty ΔT is bounded by

$$\Delta T \geq \frac{(\textit{maximal hop distance to root})}{f_C} \frac{2\hat{\rho}}{1-\hat{\rho}} \quad . \tag{2}$$

This expression illustrates that the maximal uncertainty growths with the maximal hop distance to the root. Thus, a depth-first tree yields a large maximal uncertainty.

2) Lower bound for random communications: We determine the maximal real-time difference $(t - t_s)$ and thus the maximal uncertainty for random communications by simulation. In analogy to (2), we define a *pseudo hop distance* to the root node as

$$(\textit{pseudo hop distance to root}) = f_C \cdot \max_t \{t - t_s\} \quad .$$

B. Communication frequency per node

In both the tree-based and the random-communications mode, we assume that every non-anchor node initiates a communication with some node within its transmission range with a frequency of f_C. Thus, in a time interval of size Δt, $f_C \cdot \Delta t$ communications take place per node, each involving two nodes.

In a star topology, i.e. in the ideal breadth-first tree, all child nodes initiate a communication with the root node once in an interval of length $1/f_C$. The root node hence communicates n times (where n is the number of child nodes) within each interval of length $1/f_C$. As a consequence, the root consumes n times more energy than the child nodes.[4] In a chain, i.e. in the ideal depth-first tree, the root and the last node communicate once per $1/f_C$, and all other nodes communicate twice. Thus, the energy consumption is distributed almost evenly among all nodes. In a general tree, the maximal number of communications per $1/f_C$ is equal to the maximal node degree.

In the random-communications mode, the maximal number of communications per node and time unit is determined using simulation. The communications are distributed quite uniformly among the nodes due to the random selection of communication partners. A *pseudo node degree* can be determined by simulation by counting the maximal number of communications in which any node is engaged in an interval of length $1/f_C$.

[4]Note that in our model, communication is point-to-point. If broadcast is possible, the root needs to communicate only once per $1/f_C$.

TABLE I

MAXIMAL UNCERTAINTY ΔT AND MAXIMAL COMMUNICATION FREQUENCY FOR A TRANSMISSION RANGE OF 0.3 TIMES THE AREA WIDTH AND TWO DIFFERENT CONSTANTS $\hat{\rho}$. THE UNCERTAINTY ΔT SCALES WITH $\hat{\rho}$ AND $1/f_C$.

f_C	Breadth-first tree		Depth-first tree		Random comm.	
	ΔT	comm. per node	ΔT	comm. per node	ΔT	comm. per node
Transmission range/area width $= 0.3$, $\hat{\rho} = 100$ ppm						
1/h	2.8 s	38/h	69.1 s	4/h	14.4 s	3/h
1/min	48 ms	38/min	1.2 s	4/min	240 ms	3/min
1/s	800 μs	38/s	19 ms	4/s	4 ms	3/s
Transmission range/area width $= 0.3$, $\hat{\rho} = 1$ ppm						
1/h	28 ms	38/h	691 ms	4/h	144 ms	3/h
1/min	480 μs	38/min	12 ms	4/min	2.4 ms	3/min
1/s	8 μs	38/s	190 μms	4/s	40 μs	3/s

C. Simulation study

1) Setup: We placed 100 nodes randomly in a square with edge length 100. An anchor node was placed in the center of the square. For a transmission range between 10 and 80, breadth-first and depth-first trees were constructed. Fig. 1 shows these trees for transmission ranges 15 and 35.

In these trees, the maximal hop distance of a node to the root and the maximal node degree were determined. Using the same node locations, we simulated the random-communications mode for a real-time interval of $100/f_C$. From the obtained traces, we computed pseudo hop distances and pseudo node degrees. The maximal and minimal results from 50 runs of this experiment are shown in Fig. 2.

2) Results: First, consider the maximal degree of any node in the network, shown on the right side of Fig. 2. For the depth-first tree, the maximal degree is fairly low (≤ 4) and remains approximately constant for all transmission ranges. For the breadth-first tree, the maximal degree is small for a small transmission range and then quickly increases. At a transmission range of $50\sqrt{2} \approx 71$, the root node can reach every node, and its degree is maximal, while all other nodes have degree 1.

Now, consider the maximal hop distance to the root node. At a transmission range of 10 ($x = 0.1$), there are nodes that cannot communicate with any other node, the maximal distance is therefore not shown. This occurs up to and including a transmission range of 25. For the breadth-first tree, the maximal hop distance is fairly small (≤ 4 for a transmission range of at least 30). For the depth-first tree, the maximal hop distance is considerably larger and increases quickly (≥ 83 for a transmission range of at least 30).

In the random-communications mode, the communications are distributed as evenly among the nodes as in the depth-first tree, i.e. far better than in the breadth-first-tree. The maximal hop distance (and thus the worst-case uncertainty) of the random-communications mode is much better than that of the depth-first tree, but approximately 5 times worse than that of the breadth-first tree (≤ 20 for a transmission range of at least 30).

Exemplary values for maximal uncertainty ΔT and maximal communications per node for a transmission range of 0.3 times the area width and two different drift-constraint constants $\hat{\rho}$ are given in Tab. I. The uncertainties ΔT scale with $\hat{\rho}$ and $1/f_C$, and there exist trade-offs between the three approaches.

Each row in Tab. I compares the different modes for a given *total* number of communications *in the network*. Fig. 3 shows the maximal uncertainty for a given *maximal* number of communications *per node*, which is analogous to a given minimal lifetime of each node. Here, the random-communications mode achieves strictly better results than the tree-based modes.

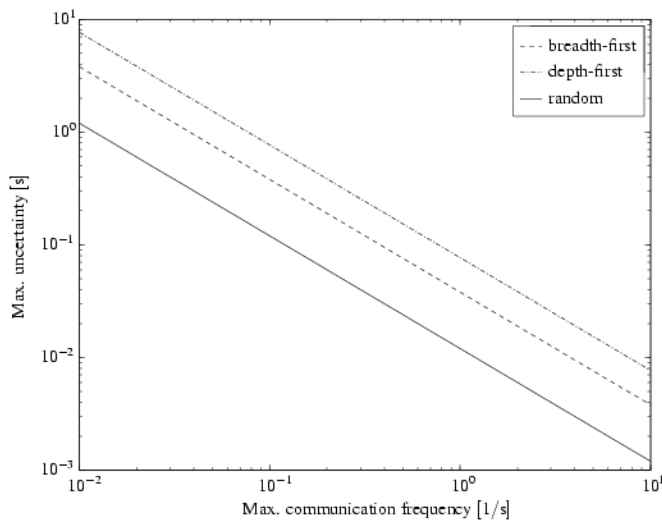

Fig. 3. For an equal maximal communication frequency per node, the random-communications mode achieves a strictly smaller uncertainty than the breadth-first and depth-first tree-based modes.

3) Discussion: In this section, we have considered networks without node mobility and shown that in comparison with the tree-based mode, the random-communications mode offers a very good compromise between a small maximal uncertainty and fair distribution of communications among the nodes. For a given maximal communication frequency per node, it achieves a strictly smaller uncertainty than the breadth-first and depth-first tree-based modes.

While the random-communications mode does not have any overhead at startup, the topology construction required in the tree-based modes is expensive. The authors of [10] state that the communication overhead of breadth-first tree construction can be reduced to $10 \cdot n \cdot m^{1/2}$, where n is the

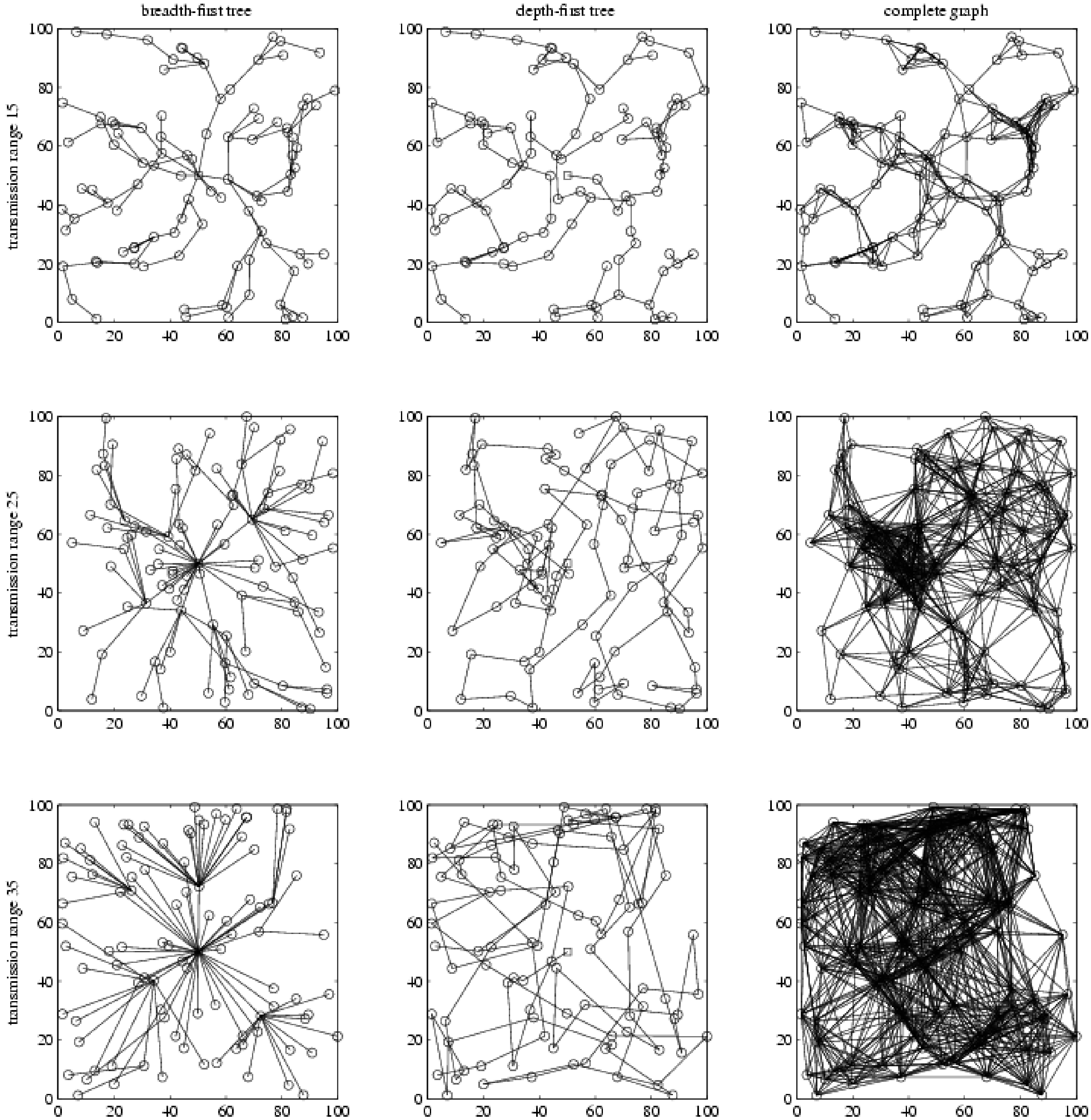

Fig. 1. 100 randomly placed nodes (depicted as circles), organized in a breadth-first tree (left column) and in a depth-first tree (middle column). The root (depicted as a square) is always in the center of the area. The right column shows the complete graph for a given transmission range.

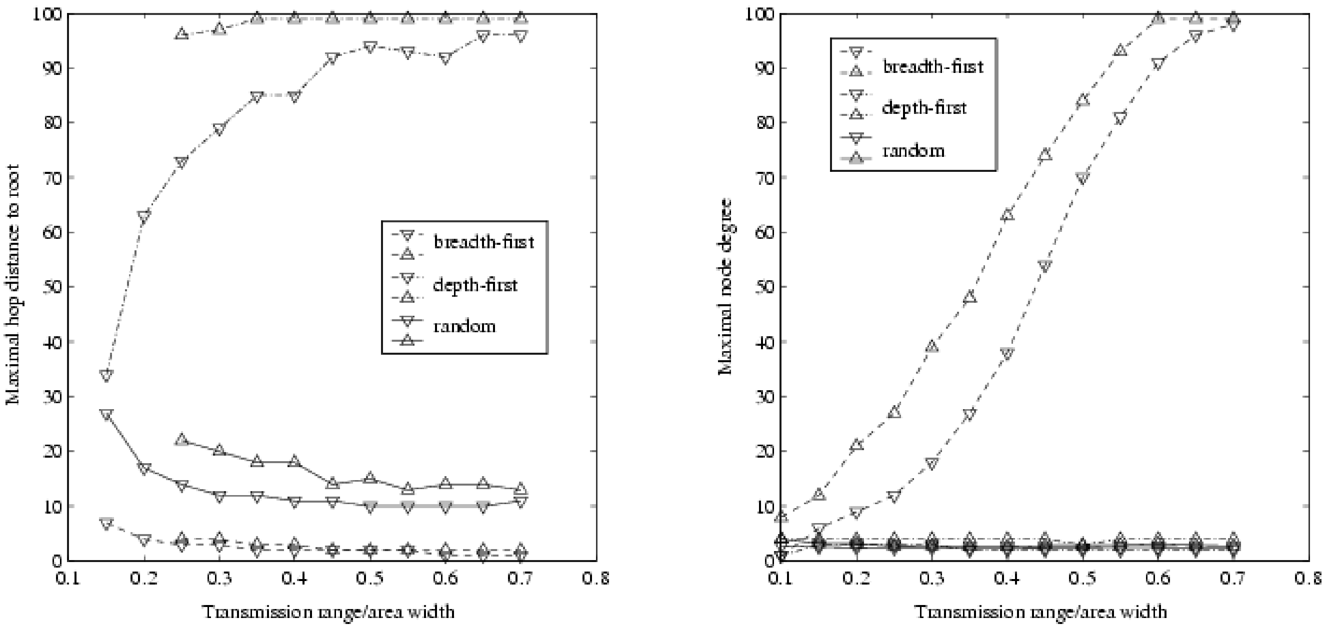

Fig. 2. Maximal hop distance to the root (left) and maximal node degree (right) of breadth-first and depth-first trees constructed in networks of 100 randomly placed nodes. The figures also show pseudo hop distances and pseudo node degrees determined from simulating the random-communications mode. The upward and downward triangles represent the maximal and minimal values from all simulation runs. The missing upward triangles on the left for transmission ranges below 30 ($x = 0.3$) are due to nodes not connected to the root.

number of nodes and m the number of edges in the network. The communication overhead of a distributed depth-first tree-construction algorithm is specified as $4 \cdot m$ in $4 \cdot n - 2$ rounds. In our simulations of the networks described above, the number of nodes is 100 and the average number of edges, assuming a transmission range of 30, is approximately 1100. Thus, the number of communications necessary for tree construction is approximately 13270 for breadth-first and 4400 for depth-first trees. This overhead becomes completely unbearable when the tree has to be reconstructed frequently due to node mobility.

IV. Mobile Nodes

We will now describe the scenarios with mobile nodes that we simulated and the performance measures we used to evaluate the results. To make our simulation results comparable to those of [5], we chose similar parameters and merely added node mobility. The simulations were done using custom C++ programs.

A. Model and Measures

We simulated sensor-network scenarios for a simulation time of 500 hours, i.e. almost 21 days. The network contained 100 nodes that were distributed uniformly at random in a square area with edge length 10000. Their local clocks had a constant drift rate between -100ppm and $+100$ppm. A number $N_A \in \{5, 10, 20\}$ of the 100 nodes were anchor nodes. In each run, all nodes had the same transmission range; between runs, it was varied between 0.1 and 0.5 times the width of the simulation area. Sensor nodes communicated with other sensor nodes at an average frequency of $f_C = 2/\text{h}$. Anchor nodes communicated with sensor nodes at a smaller average frequency $f_A = 0.02/\text{h}$.

In our mobility model, both anchor and non-anchor nodes can move according to a model similar to the random-waypoint model [20]: We first specify the number $l \geq 1$ of locations per node. For each node, we choose uniformly at random $l-1$ real times $t_1, \ldots, t_{l-1}$ within the simulation time frame and l locations $L_0, \ldots, L_{l-1}$. We set $t_0 := 0$. At real time t_i, the node is at location L_i. For any two consecutive locations i and $i+1$, the node moves with constant speed on a direct line from location i to location $i+1$. For $l > 1$, the node is in constant movement; there is no pause at locations.

The number of locations can be given separately for anchor and non-anchor nodes. If the anchor nodes have only one location, i.e. if they are immobile, then they are placed on a regular grid that maximizes their coverage of the simulation area. This is a reasonable assumption, since immobile anchor nodes can be expected to be placed optimally with respect to area coverage.

The measure for the performance of an interval-based synchronization algorithm was the time uncertainty. We evaluated it after every communication and took the average over all communications, excluding infinite time uncertainties. We thus produced the average time uncertainties for the algorithm IM and the BP-ISA. From these values, we computed the improvement of the BP-ISA. Figures 4 to 6 give all these results as percentages and in milliseconds. For every data point, 300 runs were performed. The standard deviations for the percentage improvements are shown in Tab. II.

TABLE II

STANDARD DEVIATIONS FOR THE IMPROVEMENT PERCENTAGE OF THE BP-ISA OVER ALGORITHM IM DEPICTED IN FIGURES 4 TO 6. THE VALUES FOR NO MOBILITY ($x = 1$) DIFFER CONSIDERABLY FROM THOSE FOR MOBILITY ($x > 1$).

anchors	standard deviation					
	transmission range = 500		transmission range = 1500		transmission range = 2500	
	$x = 1$	$x > 1$	$x = 1$	$x > 1$	$x = 1$	$x > 1$
5	0.96	4.63–6.16	4.29	2.32–3.04	5.88	3.14–4.08
10	1.23	3.52–4.68	3.74	1.79–2.66	4.79	2.47–3.56
20	1.86	2.75–4.15	3.39	1.34–2.48	4.21	1.90–2.78

B. Discussion

We draw the following conclusions from Figures 4 to 6:

1) The uncertainties become smaller when the mobility becomes larger (for $x > 1$). This effect diminishes with increasing transmission range, since the improvement in connectivity becomes smaller.
2) Analogously, the uncertainties become smaller when the transmission range increases. This effect is larger for low mobility than for high mobility.
3) For transmission range 500, there is a sharp jump between $x = 1$ and $x = 5$ in the graphs showing the uncertainty, and the uncertainty for $x = 1$ is much smaller than that for $x = 5$, suggesting that no mobility leads to smaller uncertainty. The reason for this is that at $x = 1$, i.e. without mobility, most of the nodes are not connected to the anchor. Their uncertainty thus is infinite and is not considered. The few nodes that are connected to the anchor are only one hop away from it. Their uncertainty thus is very small. As the transmission range grows larger, this effect diminishes.

V. CONCLUSION

In this paper, we extended the results from [5] by showing that node mobility puts no burden on interval-based synchronization, but actually increases its quality. Furthermore, we argued that synchronization algorithms that employ hierarchical communication structures (e.g., [8], [9], [10], [19]) are bound to suffer from prohibitive overhead if nodes are mobile. We showed that for the interval-based approach, unstructured communication offers a very good compromise between a small maximal uncertainty and fair distribution of communications among the nodes.

Future work consists in exploring different mobility models, repeating our simulations for broadcast instead of point-to-point communication, and refining the simple bounded-drift clock model. By including bounds on drift variation, we expect to be able to reduce the uncertainty considerably.

REFERENCES

[1] K. Römer, “Time synchronization in ad hoc networks,” in *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*. ACM Press, Oct. 2001, pp. 173–182.

[2] IEEE Computer Society LAN MAN Standards Committee, *IEEE 802.11: Wireless LAN Medium Access Control and Physical Layer Specifications*, Aug. 1999.

[3] H. Woesner, J.-P. Ebert, M. Schlager, and A. Wolisz, “Power saving mechanisms in emerging standards for wireless LANs: The MAC level perspective,” *IEEE Personal Communications*, vol. 5, no. 3, pp. 40–48, June 1998.

[4] D. L. Mills, “Internet time synchronization: The network time protocol,” *IEEE Transactions on Communications*, vol. 39, no. 10, pp. 1482–1493, Oct. 1991.

[5] P. Blum, L. Meier, and L. Thiele, “Improved interval-based clock synchronization in sensor networks,” in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN ’04)*, Apr. 2004, pp. 349–358.

[6] L. Meier, P. Blum, and L. Thiele, “Internal synchronization of drift-constraint clocks in ad-hoc sensor networks,” in *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2004)*, May 2004, pp. 90–97.

[7] J. Elson, L. Girod, and D. Estrin, “Fine-grained network time synchronization using reference broadcasts,” *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 147–163, 2002.

[8] S. Ganeriwal, R. Kumar, and M. B. Srivastava, “Timing-sync protocol for sensor networks,” in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys ’03)*. ACM Press, 2003, pp. 138–149.

[9] M. L. Sichitiu and C. Veerarittiphan, “Simple, accurate time synchronization for wireless sensor networks,” in *IEEE Wireless Communications and Networking (WCNC 2003)*, vol. 2. IEEE, 2003, pp. 1266–1273.

[10] J. van Greunen and J. Rabaey, “Lightweight time synchronization for sensor networks,” in *Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications (WSNA ’03)*. ACM Press, 2003, pp. 11–19.

[11] D. L. Mills, “Bibliography on computer network time synchronization,” available on-line at http://www.eecis.udel.edu/˜mills/bib.html.

[12] B. Simons, J. L. Welch, and N. A. Lynch, “An overview of clock synchronization,” in *Fault-Tolerant Distributed Computing*, ser. Lecture Notes in Computer Science, B. B. Simons and A. Z. Spector, Eds., vol. 448. Springer, 1990, pp. 84–96.

[13] F. Sivrikaya and B. Yener, “Time synchronization in sensor networks: A survey,” *IEEE Network*, vol. 18, no. 4, pp. 45–50, July 2004.

[14] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, “Clock synchronization for wireless sensor networks: a survey,” *Ad Hoc Networks*, vol. 3, no. 3, pp. 281–323, 2005.

[15] K. Römer, P. Blum, and L. Meier, “Time synchronization and calibration in wireless sensor networks,” in *Handbook of Sensor Networks: Algorithms and Architectures*. John Wiley & Sons, Sept. 2005.

[16] K. Marzullo and S. Owicki, “Maintaining the time in a distributed system,” in *Proceedings of the 2nd annual ACM Symposium on Principles of Distributed Computing*. ACM Press, 1983, pp. 295–305.

[17] U. Schmid and K. Schossmaier, “Interval-based clock synchronization,” *Real-Time Systems*, vol. 12, no. 2, pp. 173–228, 1997.

[18] G. J. Pottie and W. J. Kaiser, “Wireless integrated network sensors,” *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, May 2000.

[19] M. Maróti, B. Kusy, G. Simon, and A. Lédeczi, “The flooding time synchronization protocol,” in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys’04)*. ACM Press, 2004, pp. 39–49.

[20] D. B. Johnson and D. A. Maltz, “Dynamic source routing in ad hoc wireless networks,” in *Mobile Computing*, Imielinski and Korth, Eds. Kluwer Academic Publishers, 1996, vol. 353.

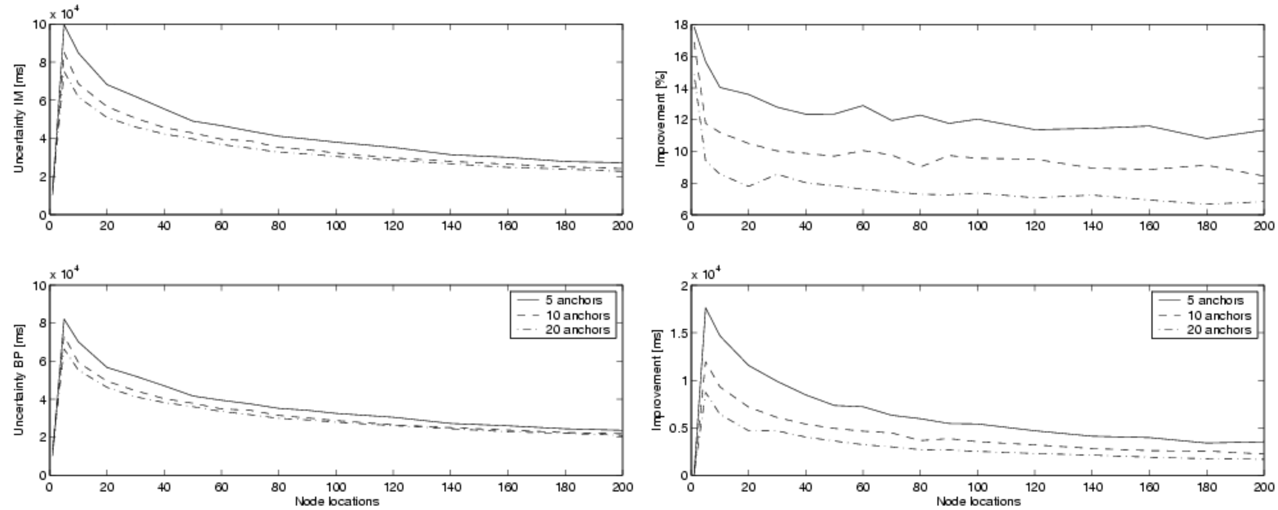

Fig. 4. Simulation results for transmission range 500.

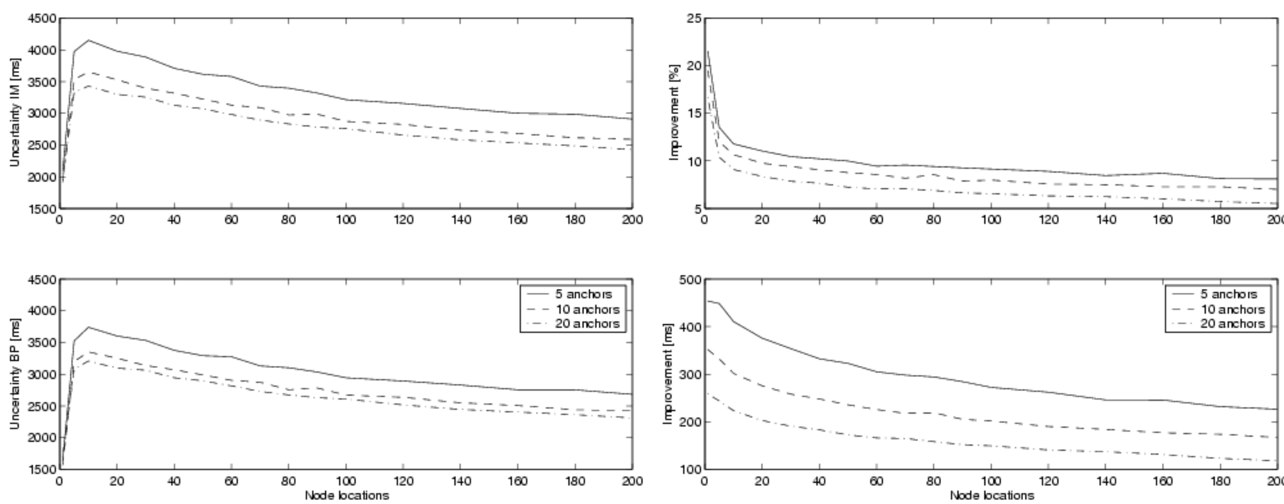

Fig. 5. Simulation results for transmission range 1500.

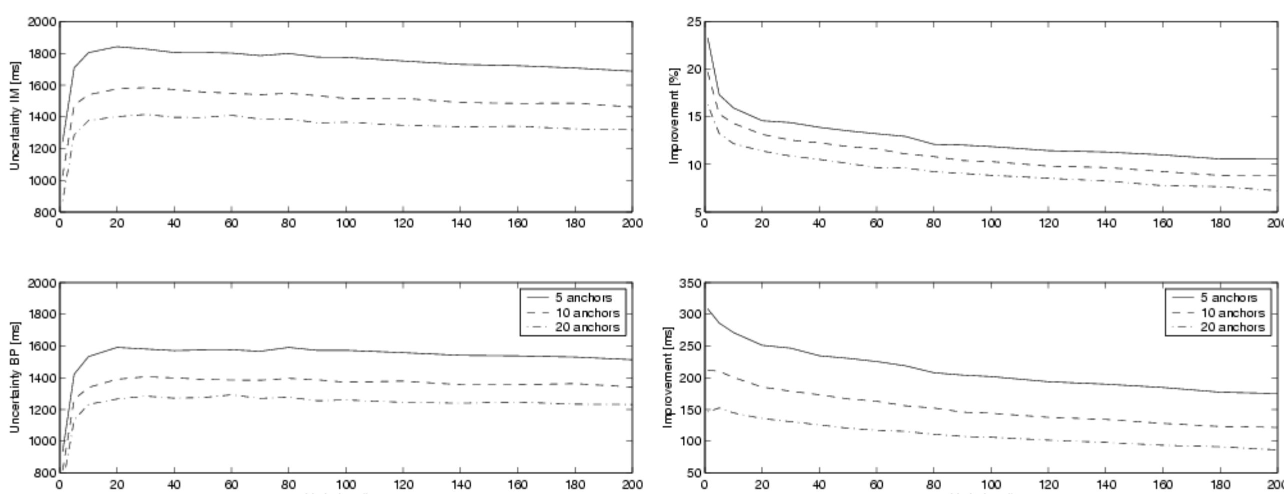

Fig. 6. Simulation results for transmission range 2500.