Directed Budget-Based Clustering for Wireless Sensor Networks

Leonidas Tzevelekas
National and Kapodistrian
University of Athens
Email: ltzev@di.uoa.gr

Ioannis Stavrakakis
National and Kapodistrian
University of Athens
Email: ioannis@di.uoa.gr

Abstract— **Wireless Sensor Networks are typically bound to operate autonomously on a field, under severe energy constraints and without any centralized control. It is thus essential to develop self-organization protocols/algorithms which enable the autonomous, distributed and energy efficient network self-organization. Budget-based clustering approaches have recently been proposed for this purpose, by specifying rules for distributing a given budget of tokens to neighbors. In this paper, two strictly localized, budget-based clustering algorithms are proposed: the Directed Budget-Based (DBB) and Directed Budget-Based with Random Delays (DBB-RD). The basic, innovative idea is to utilize clustering status information that can be readily available (e.g. through the HELLO exchanges) to reduce or eliminate token distribution contentions (both intra- and inter-cluster) that severely limit the effectiveness of earlier budget-based approaches. Simulation results are presented demonstrating a substantial improvement over the earlier approaches with respect to the achieved clustersizes and time to complete network decomposition.**

I. Introduction

Wireless Sensor Networks (WSNs) typically consist of large numbers of individual sensor nodes which are tiny, compact and cheap embedded devices. The sensor nodes are equipped with low-range wireless transceivers enabling them to communicate with their local neighbors and ultimately with the rest of the network. Despite their limited hardware and energy resources, the collective behavior of the sensor network can be arbitrarily complex and sophisticated.

Recently, the emergence of WSNs as a promising new platform for networking and processing of sensed data renewed the interest in designing clustering algorithms which can be applied to sensor networks. The novel, highly constrained hardware characteristics of individual sensor nodes and the completely new networking applications envisioned for WSNs call for increasingly distributed, localized and primarily energy/message efficient algorithms for clustering.

The existence of clusters of nodes in WSNs can enhance the network operation as demonstrated in recent works [1], [2]. Such a network decomposition can vividly enhance sensor node coordination, network management and in-network processing and aggregation of sensor data. The clustered network architecture is typically advantageous to be based on clusters of fixed size (reduced routing protocol overhead, better accommodating specific service requirements, etc). At the same time, it is very important that the self-organization process does not drain the batteries of the sensor nodes [3]. It is thus crucial that self-organization algorithms not only yield clusters of sizes close to the fixed targeted value, but also complete the network self-organization process within a short time, or by executing a low number of steps or requiring low message exchanges.

Another constraint to consider when designing algorithms for WSNs is due to the fact that each individual sensor node is typically bound to communicating with its immediate neighbors only. Long-haul communications in WSNs are not desired or possible due to severe energy constraints as well as the environmental factors that influence the sensor's communication and networking capabilities. Consequently, it is important to design algorithms and protocols for WSNs which are *localized* or even *strictly localized*, as explained in [4]. A *strictly localized protocol* is a localized protocol in which all information processed by a node is either: (a) local in nature or (b) global in nature, but obtainable by querying only the node's neighbors or itself [5]. The local interactions are primarily enabled through the exchange of periodic local HELLO messages, which account for building and maintaining the local neighborhood and may also carry any information needed to achieve the desired global objectives.

Two algorithms for message efficient distributed clustering are proposed in [3]. The goal is to decompose an autonomous sensor network into clusters of bounded size while keeping the overall message complexity low. These algorithms are supplemented by a randomized technique for specifying the clustering initiation process.

The focus of the aforementioned algorithms is to design distributed, energy efficient algorithms for network organization. Recently there has been a focus on designing localized algorithms for various network functions, such as flooding, broadcasting and spanner construction [6], [7]. These algorithms utilize the existing periodic HELLO message exchanges to maintain and update the local neighborhood of each node (through HELLO exchanges with first-hop neighbors) and to build the desired structure over it.

In this work, a new *strictly localized* clustering protocol

This work has been supported in part by the Greek Secreteriat for Research and Technology under the research Program PENED 2003 and the European Union under the programs E-NEXT NoE (IST-506869) and ANA (Autonomic Network Architecture, IST-FET 27489)

is proposed for WSNs which aims at decomposing large scale sensor networks into non-overlapping clusters of fixed size. The primary focus of this work is to provide a fast network decomposition algorithm, which constructs clusters with sizes (in number of nodes) as close as possible to a desired upper bound. The proposed algorithm is considered to be executed in rounds that coincide with the periodic exchanges of HELLO messages and takes advantage of their presence in order to convey (with minimal additional overhead through a flag) some elementary and limited clustering process status information.

II. Distributed Budget-based Clustering

In this section some definitions to be used in this paper are introduced along with some background information on budget-based clustering.

The network of nodes and edges are viewed as an undirected graph $G = (V, E)$ on the plane, where V represents the set of nodes with $|V| = n$ nodes and E represents the set of edges with $|E| = m$ edges. An edge $(u, v) \in E$ exists iff nodes $u, v \in V$ can directly communicate with each other. Each node $v \in V$ has a unique identity ID(v). A *cluster* is a connected component of the original graph G together with a node designated as the initiator node $v_0 \in V$ and a spanning tree rooted at the initiator node. The maximum allowed size for the clusters is called the *cluster size bound* B, where $B \ll n$.

The proposed work falls into the category of strictly localized protocols, since all the required information for the protocol to run at each node is obtainable from the immediate, local neighbors of that node. There is no communication between nodes that are more than one hop away at any stage of execution of the proposed clustering protocol.

Budget-based, bounded size, message-efficient, distributed algorithms for self-organization in WSNs are proposed in [3]. The main idea of the algorithms is that nodes allocate local (cluster) growth budgets to their neighbors, starting with an initiator node and a specific budget of tokens. The nodes then allow the cluster to grow based on local decisions rather than involving the initiator at each round. As a result, the number of messages exchanged is significantly reduced when compared with older and more commonly used approaches, such as the *Expanding Ring* approach [8].

Two algorithms for message-efficient, budget-based clustering in WSNs are proposed in [3]: the *Rapid* and the *Persistent* algorithms. The Rapid algorithm is a fast algorithm for producing clusters of bounded size. The Persistent algorithm is the recursive elaboration of Rapid aiming at improving the poor worst case performance of Rapid. The most important characteristic of both algorithms is the budget distribution process as a method for appending additional nodes to the currently growing cluster. Both algorithms are initiated by a random network node (starting the budget distribution process) referred to as the *initiator* node.

In Rapid, the initiator node is assigned a budget of B tokens of which it accounts one for itself (and this makes it a member of the cluster) and distributes *evenly* $B - 1$ tokens among its neighbors. The neighbors that receive the token(s), account one for themselves and distribute equally the remaining among all their neighbors except the parent node. The same procedure executes for all children nodes of the initiator until the budget is exhausted. Each node receiving a message sends an acknowledgment to its parent when either the budget is exhausted or when it has received an acknowledgement from all its contacted children-nodes. The algorithm terminates when the initiator node receives acknowledgements from all the neighbors contacted.

The Persistent algorithm on the contrary, does not immediately send acknowledgement to the parent node when it receives acknowledgements from all the contacted nodes. Instead, it calculates the shortfall between assigned budgets and subtree sizes replied for all the contacted nodes. If the shortfall is nonzero, the algorithm tries to re-distribute the shortfall evenly among the nodes not already contacted and the nodes that successfully utilized/distributed their previous budgets. The algorithm terminates when the tokens are exhausted or when no more growth is possible for the cluster.

An integral component of a budget-based clustering scheme is a mechanism for selecting initiator nodes. A randomized initiator selection scheme is proposed in [3] under which a node decides to initiate a cluster after a carefully selected exponentially distributed time. Caution is taken to avoid having two clusters compete for the same nodes, by probabilistically assuring that any two growing clusters will be sufficiently far apart both in time and in space and have enough room to grow. In the simulations the sequential approach is used to fire an initiator as soon as the previously growing cluster is completed. This approximation of the network decomposition into clusters is proved to be sufficiently good for measuring the algorithm's clustering performance and not far from the real world performance under an effective design for the initiators' firing.

The Rapid algorithm has low message complexity $O(B)$ but has poor worst-case performance in terms of the cluster sizes produced. The Persistent algorithm, being the recursive elaboration of Rapid, has excellent clustering performance in terms of achieved cluster sizes, but lacks in performance in terms of message complexity $O(B^2)$ or in required time until full network decomposition.

A drawback in the aforementioned algorithms is that they "blindly" distribute budget to nodes, including nodes that are likely to waste them (Rapid) or return them (Persistent), impacting negatively on the cluster size or time to network decomposition /message efficiency.

III. The proposed clustering algorithms

In this section two strictly localized, budget-based clustering algorithms are described: the Directed Budget-Based (DBB) algorithm and the Directed Budget-Based with Random Delays algorithm.

A. *The Directed Budget-based (DBB) clustering algorithm*

As mentioned earlier the Rapid and Persistent algorithms distribute tokens blindly in the sense that the state of the neighbor (i.e., whether already clustered or not) is not taken into consideration. Consequently, tokens are wasted (or returned for Persistent) when distributed to nodes which are already clustered by another initiator. This blindness is the reason for their poor clustering performance in terms of cluster sizes (Rapid) or time to network decomposition completion (Persistent).

To reduce the inefficiencies due to the blindness of the token distribution process, it is proposed in this paper that nodes update their neighbors regarding their clustering status as described next. This low overhead status exchange (see next) will enhance the effectiveness of the token distribution process, as nodes will utilize this status information in order to direct tokens towards areas of (yet) unclustered neighboring nodes and reduce token waste.

The proposed clustering algorithm will be described in steps that will be initiated with the activation of an initiator node and will be completed when the associated cluster is completed. The number of steps (rounds) required for creating a cluster will be one measure of the efficiency of the algorithms; such a metric also captures (in general) the level of message exchange complexity in addition to measuring the time to network decomposition (assuming that the time interval between two consecutive steps is fixed).

Without loss of generality it will be assumed in the remaining of the paper that the clustering algorithm's steps are executed after each periodic exchange of HELLO messages. That is, the associated time between consecutive steps will be defined by the period of the HELLO message process and the clustering status information (clustered or not clustered - a 0/1 flag) will be embedded on these messages. The motivation behind this (otherwise non restrictive) proposal is to utilize message exchanges typically taking place in a WSN (as part of the needed topology discovery and maintenance process) and avoid introducing additional message exchanges between energy constrained nodes. The HELLO message exchange process is always present for topology discovery or maintenance and this is the time when cluster formation or re-organization needs to take place as well.

As it is well known, the standard HELLO messages reveal to the nodes their *physical* neighborhood and collectively these physical neighborhoods determine the physical topology. By adding a simple flag in the HELLO messages indicating whether the sending node is clustered or not, the *unclustered* neighborhood is revealed to the nodes and collectively these unclustered neighborhoods determine the unclustered topology; the unclustered neighborhood is defined as the subset of the physical neighborhood (set of neighbors) that is not associated with any cluster.

The HELLO messages may also be utilized to carry the clustering messages (token distribution messages and acknowledgement messages). Specifically, it is assumed that a node will piggyback a clustering message in the (anyway sent as part of the topology update procedure) HELLO message. The execution of the proposed DBB algorithm may therefore be completely embedded into the periodic topology updating and maintenance procedures of the WSN.

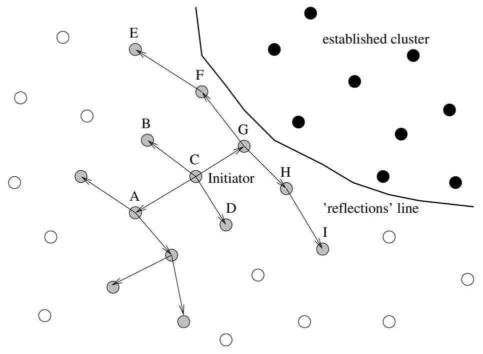

Fig. 1. Directed Budget-Based Clustering

In the proposed Directed Budget-Based (DBB) algorithm, the unclustered neighborhood is identified for each node after each HELLO message exchange and tokens are distributed (equally) over this unclustered neighborhood. Compared with an algorithm distributing tokens over the physical topology (e.g. the Rapid algorithm in [3]), the proposed DBB algorithm directs the token distribution process away from clustered regions by operating on the unclustered (as opposed to the physical) topology. The clustered and unclustered topologies typically consist of multiple island-regions scattered throughout the network. Each island-region of clustered nodes may be viewed as defining a boundary that bounces (or reflects) away any incoming (and bound to be wasted or returned) tokens and directs the clustering (token distribution) process towards unclustered regions.

Fig. 1 illustrates the spatial evolution of the clustering process initiated from node C. The unclustered topology (an island-region) is depicted by the grey nodes and the clustered by the black nodes. The boundary of the clustered island region shown prevents the tokens distributed by nodes G, H, F, etc., from entering this region and, thus, be wasted or returned; in other words, tokens are directed away from this region or are bounced away on the boundary of this region.

For the network decomposition scenario under sequentially firing initiators (as described earlier), the DBB algorithm eliminates the inter-cluster token distribution contentions, i.e., wastes (or returns) of tokens due to directing the propagating tokens of a growing cluster towards an already clustered region of the network. In the more general scenario under which multiple initiators can be concurrently active in the network, though, some inter-cluster token distribution contentions may not be avoided, since a node may receive tokens from two simultaneously growing clusters in its neighborhood. In this case the proposed DBB algorithm, while still significantly reducing inter-cluster token distribution contentions, it may not eliminate them completely.

As indicated earlier, the performance of the proposed algorithm will be evaluated in terms of (a) the deviation of the achieved mean cluster size from the targeted bound B and (b) the mean time to network decomposition. The underlying assumption here is that the HELLO message exchanges already exist in real environments and, thus, they do not constitute an overhead introduced by the clustering algorithm, beyond the additional 1-bit flag and recording this information on a table.

B. The Directed Budget-Based Algorithm with Random Delays (DBB-RD)

The proposed DBB algorithm saves tokens and is expected to enhance the clustering process by reducing or eliminating inter-cluster token distribution contentions. Tokens are also likely to be wasted due to intra-cluster token distribution contentions arising when two nodes have common (first-hop) neighbors. The high node densities typically observed in WSNs result in many nodes having common (first-hop) neighbors in their local neighborhood. Thus, when two such nodes take part in the growing of a cluster under a given initiator (i.e. they both have part of the budget to distribute further), it is likely that they will pick the same common neighbor (or neighbors) to forward part of their tokens to.

The intra-cluster token distribution process contentions for Rapid, Persistent and DBB are due to the fact that these algorithms proceed with the budget distribution at each node immediately upon receiving a budget from one of their neighbors. However, the overall clustering performance of the DBB algorithm would be enhanced if, for example, each distributing node were aware that some of its neighbors were already clustered due to receiving tokens from the local neighborhood and diverted its tokens towards another unclustered node, even if that required that it waited for a later round of HELLO message exchanges to complete its budget distribution.

The DBB algorithm with Random Delays is presented here as a modified version of the DBB algorithm aiming to reduce primarily intra- but also inter- cluster token distribution contentions. The basic idea is to delay the token forwarding to a neighbor by a random number of rounds (or HELLO exchanges) to reduce the probability of token distribution contention (notice that this Random Delay is zero for the DBB algorithm). This way, the neighboring, distributing nodes will likely be aware of earlier budget distribution effects on the local neighborhood (which nodes are clustered/not clustered) and thus divert their tokens towards unclustered local neighbors only. The cost associated with the introduction of Random Delays in the DBB algorithm is an additional delay in the overall time until the network decomposition is completed. This additional delay results in "de-synchronizing" the execution of the potentially synchronized token distribution process at neighboring nodes and reduce the likelihood for collisions. More details on the relative performance improvement and the associated trade offs are presented in the next section.

IV. SIMULATION RESULTS

The Rapid, Persistent, DBB and DBB-RD algorithms are all implemented in a C++ based simulator (Omnet++). The simulated network consists of $n = 6000$ nodes, which are randomly placed on a square plane with size $l = 1000\ m$. Each node's (x, y) coordinate is randomly drawn using a uniformly distributed random variable in the range $[0, l)$. The individual sensor's wireless communication range is fixed to $r = 25\ m$, which is a typical value for the low-range communication capabilities of a sensor node. These WSN characteristics result in an average network density of $0.006\ nodes/m^2$, which, if multiplied by each sensor's coverage disc, leads to an average connectivity degree of $\rho = 11.781$ nodes. The nodes in the network are checked for network connectivity and disconnected ones are "moved" randomly within the connectivity region of the closest connected neighbor to ensure connectivity of the entire network graph.

The primary objective of the simulation experiments is to study comparatively the clustering performance of the Rapid, Persistent, DBB and DBB-RD algorithms. The sequential initiators approach is adopted in the network (as described in section II); i.e., it is assumed that there is only *one cluster growing* in the entire network at a time.

The clustering performance of the described algorithms, is evaluated with respect to the following metrics: (a) The total number of clusters in the network after completion of the network decomposition; (b) The average cluster size of all clusters created; (c) The number of consecutive execution rounds required until the final cluster is completed. The ideal number of clusters to be formed, I, depends on the selected clusterbound B and it is given by $I = \lfloor n/B \rfloor$. The performance of the algorithm increases the closer the induced number of clusters to I, the closer the induced mean cluster size to B and the smaller the number of execution rounds.

For each given set of simulation parameters, $K = 5$ independent runs were carried out. The sample mean and standard deviation of the measured quantity x are calculated from $\overline{x} = \frac{1}{K}\sum_i^K x_i$ and $s = \sqrt{\frac{1}{K-1}\sum_i^K (x_i - \overline{x})^2}$, respectively, and under the normality assumption for the mean η of a calculated metric, it is shown in [9] that $\eta = \overline{x} \pm t_{1-\delta/2}\frac{s}{\sqrt{n}}$, where $t_{1-\delta/2} = t_u$ is the u-percentile of the t-Student distribution with $K - 1 = 4$ degrees of freedom in this case. The confidence intervals calculated and presented next use a confidence coefficient of 0.95, or in other words the 0.95-confidence intervals are estimated.

A. Results for the Rapid & Persistent Algorithms

The simulation results of the clustering algorithms are obtained for two different clusterbound values: $B = 30$ and $B = 60$. These particular choices for B are considered as reasonable clusterbounds for medium- and large- sized clusters. Since the average connectivity degree of the WSN nodes is approximately 11.5, tokens will need to be distributed beyond the initiator's local neighbors and avoid the degenerate case of network decomposition consisted of one-hop clusters around the initiator.

Table I presents results for network decomposition using the Rapid/Persistent algorithms with sequential picking of initiators. One can clearly see that for Rapid the mean cluster size is very low compared with the assigned clusterbound for the clusters in the network (8.69 for $B = 30$ case and 11.13 for $B = 60$ case). This may be attributed to the fact that many tokens are wasted during the budget distributions in

TABLE I

Results for Rapid & Persistent

	Average Number of Clusters	**Average Clustersize**	**Average Number of Rounds**
($B = 30$)			
Rapid	691 ± 6.9	8.69 ± 0.087	3258 ± 29.2
Persistent	286 ± 13.3	21.03 ± 0.98	18992 ± 875.9
($B = 60$)			
Rapid	539 ± 14.3	11.13 ± 0.297	2610 ± 65.4
Persistent	158 ± 8.8	37.99 ± 2.111	18548 ± 735.5

TABLE II

Results for DBB & DBB-RD

	Average Number of Clusters	**Average Clustersize**	**Average Number of Rounds**
($B = 30$)			
DBB (no Delay)	539 ± 7.8	11.12 ± 0.163	2558 ± 22.2
DBB-RD ($r = 3$)	422 ± 11.2	14.22 ± 0.381	3247 ± 56.8
DBB-RD ($r = 5$)	402 ± 11.7	14.92 ± 0.435	3965 ± 33.2
DBB-RD ($r = 10$)	386 ± 2.6	15.53 ± 0.1	5868 ± 53.7
($B = 60$)			
DBB (no Delay)	426 ± 7.3	14.1 ± 0.246	2167 ± 46
DBB-RD ($r = 3$)	308 ± 11.8	19.52 ± 0.728	2933 ± 96.8
DBB-RD ($r = 5$)	293 ± 10	20.46 ± 0.687	3569 ± 49
DBB-RD ($r = 10$)	282 ± 3.9	21.31 ± 0.3	5329 ± 112.1

Rapid, due to the 'blindness' of the algorithm with respect to already clustered nodes in the network (intra- and inter-cluster token distribution collisions) and thus many small-sized clusters are formed in the network. On the other hand, Rapid appears to be a fast network decomposition algorithm, when compared to the significant number of operation rounds required by Persistent to complete the network decomposition (e.g. 3258 rounds required by Rapid compared with 18992 rounds required by Persistent when $B = 30$).

The Persistent algorithm is extremely slow and involves significantly more rounds of operation to complete (up to 6 times more than Rapid), primarily due to its recursive nature. However, the repeated redistribution of the unused tokens results in utilizing more unclustered nodes residing in the area around the growing cluster and an improved clustering performance (mean clustersize of 21 for the $B = 30$ and of 38 for the $B = 60$). The mean number of clusters finalized in the network is closer to the optimal ones (285.6 compared to the optimal number 200 ($B = 30$) and 158.2 compared to the optimal of 100 clusters ($B = 60$)). Note that the desired clusterbound, is far from being reached by Rapid, while Persistent improves it at the expense of increased network decomposition time and also higher message complexity.

B. Results for the proposed DBB and DBB-RD Algorithms

Table II presents similar results for the DBB and DBB-RD algorithms. Note that the results of DBB are listed as a special case of DBB-RD with zero Random Delay.

The DBB algorithm performs better than the Rapid in terms of average clustersize achieved (or number of clusters finalized) and in the overall time to network decomposition. The bouncing of tokens on the established cluster regions leads to fewer token wastes and thus to higher average clustersizes. For $B = 30$, the mean clustersize is approximately 11.1 (an increase of 28% compared to Rapid), whereas for $B = 60$ it is 14.1 (an increase of 26.6% compared to Rapid). The overall time to network decomposition is reduced compared to Rapid by a significant amount. One would expect that DBB would be faster than Persistent (due to the lack of recurrences), but the results indicate that it is also faster than the Rapid. This can be attributed to the better utilization of the distributed tokens leading to more nodes being appended to clusters at each round, and thus faster overall network decomposition.

Table II also presents results for the DBB-RD algorithm with $B = 30$ and $B = 60$ and for the random delay factor in the range $[0, r-1)$ for $r = 3$, $r = 5$ and $r = 10$. As expected, a greater range of values for the random delay parameter r leads to more time (or rounds of operation) until the network becomes fully decomposed, whereas at the same time the algorithm performs better in terms of clustering characteristics (average clustersizes achieved, average number of clusters) since token distribution contentions are less likely.

Fig. 2 presents an illustration of the algorithm's behavior (in terms of total number of rounds required until network is decomposed) when the random delay parameter increases. Note here that the case $r = 0$ represents the DBB algorithm. It is evident from the plot that the increasing random delays of DBB-RD affect the overall network decomposition time negatively. However the increased overall delays represent a trade-off for better clustering performance, as is shown in Fig. 3. One can observe that, while the overall network decomposition delay grows almost linearly with r, the increase in the average clustersizes achieved does not have the same linear behavior. Instead, there is a sharp increase in clustering performance after the introduction of the randomization parameter (for $r = 3$ the relative increase in average clustersize achieved is 27.85% for $B = 30$ and 38.44% for $B = 60$ when compared with $r = 0$) while the increase in performance for further values of the delay parameter r is not that steep anymore (only 4.88% relative increase for $B = 30$ and 4.82% for $B = 60$, when r increases from $r = 3$ to $r = 5$). This behavior of the DBB-RD algorithm suggests that the introduction of the randomization parameter enhances the quality (clustersizes achieved) of the network decomposition; however, the cost in overall delay associated with it should be carefully considered in order to achieve the best trade off between increased clustering performance and additional overall delay that can

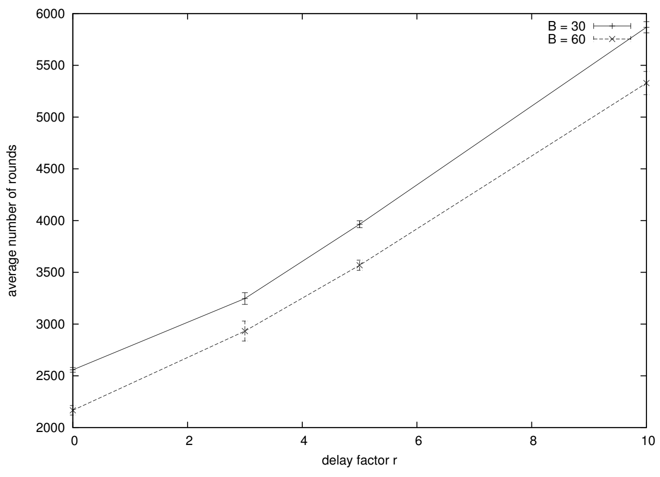

Fig. 2. Network decomposition time

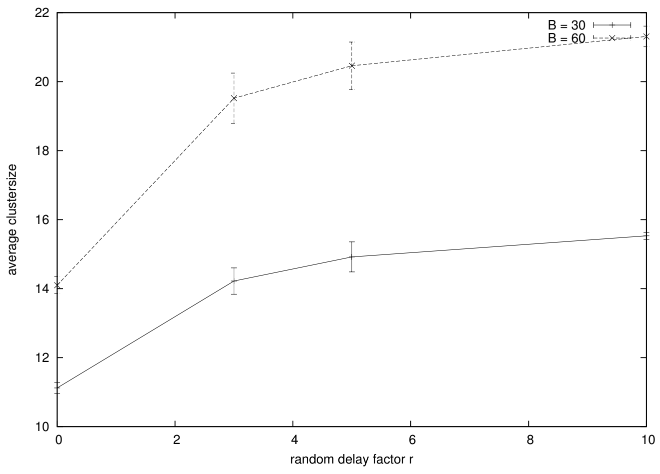

Fig. 3. Average clustersize as a function of the random delay

be tolerated.

Regarding the relative performance between the DBB-RD algorithm and the Rapid algorithms consider the case for $B = 30$ for the Rapid algorithm and for $B = 30$, $r = 3$ for the DBB-RD algorithm. Note that the two algorithms require approximately the same amount of rounds of HELLO message exchanges to complete (3258 rounds for Rapid compared to 3246.8 rounds for DBB-RD), whereas the achieved average clustersizes are 8.6878 for Rapid and 14.2228 for DBB-RD. That is a 63.71% increase in the average clustersize achieved for the proposed DBB-RD algorithm.

This tremendous improvement in clustering performance of the DBB-RD is due to the fact that it successfully deals with the sources of token wastes throughout the network decomposition, while involving low overhead (only a 1-bit variable in the HELLOs) for the network. Similar performance improvements are obtained for the case of $B = 60$, where the average number of rounds required to complete the decomposition is 2609.9 for Rapid and 2933 for DBB-RD, while the average clustersize achieved is increased by 75.29%.

V. Conclusions

In this paper, two new, strictly localized algorithms for distributed, directed, budget-based clustering of large-scale WSNs are proposed: the Directed Budget-Based (DBB) and the Directed Budget-Based with Random Delays (DBB-RD) algorithm. The basic, innovative idea is to utilize clustering status information that can be readily available to reduce or eliminate token distribution contentions (both intra- and inter- cluster) that severely limit the effectiveness of earlier budget-based approaches. The algorithms take advantage of the periodic exchanges of standard HELLO messages between neighbor nodes (apparent in real-world WSNs) to update their physical, as well as unclustered topology of their local neighborhood. They use the updated topology to direct distributing tokens away from already clustered nodes (both intra- and inter- clustered), thus significantly improving the overall clustering performance. Furthermore, the algorithms involve only moderate overhead (a simple 0/1 flag at the end of each HELLO message) to the network. Simulation results are derived showing that the new, strictly localized algorithms outperform older approaches with respect to both the clustersizes achieved and the time required (or rounds of HELLO exchanges) to complete the decomposition of the network.

References

[1] S. Bandyopadhyay and E. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *Proc. of IEEE INFOCOM'03*, 2003.

[2] O. Younis and S. Fahmy, "Distributed clustering in adhoc sensor networks," in *Proc. of IEEE INFOCOM'04*, 2004.

[3] R. Krishnan and D. Starobinski, "Efficient clustering algorithms for self-organizing wireless sensor networks," *Ad Hoc Networks*, vol. 4, no. 1, pp. 36–59, January 2006.

[4] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," in *Proc. of ACM MobiComm'99*, August 1999, pp. 263–270.

[5] H. Chan and A. Perrig, "Ace: An emergent algorithm for highly uniform cluster formation," in *Proc. of IEEE European Workshop on Wireless Sensor Networks (EWSN'04)*. Springer Verlag, January 2004, pp. 154–171.

[6] L. Orecchia, A. Panconesi, C. Petrioli, and A. Vitaletti, "Localized techniques for broadcasting in wireless sensor networks," in *Proc. of ACM DIALM-POMC'04*, Philadelphia, USA, October 2004.

[7] Y. C. Tseng, S. Y. Ni, and E. Y. Shih, "Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network," *IEEE Trans. Comput.*, vol. 52, no. 5, May 2003.

[8] C. V. Ramamoorthy, A. Bhide, and J. Srivastava, "Reliable clustering techniques for large, mobile packet radio networks," in *Proc. of IEEE INFOCOM'87*, San Francisco, USA, March 1987.

[9] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*, 4th ed. Mc Graw Hill, 2002.