Improving the Energy Efficiency of Directed Diffusion Using Passive Clustering⋆

Vlado Handziski, Andreas Köpke, Holger Karl,
Christian Frank, and Witold Drytkiewicz

Telecommunication Networks Group, Technische Universität Berlin
Sekr. FT 5, Einsteinufer 25
10587 Berlin, Germany
{handzisk,koepke,karl,chfrank,drytkiew}@ft.ee.tu-berlin.de
http://www.tkn.tu-berlin.de

Abstract. Directed diffusion is a prominent example of data-centric routing based on application layer data and purely local interactions. In its functioning it relies heavily on network-wide flooding which is an expensive operation, specifically with respect to the scarce energy resources of nodes in wireless sensor networks (WSNs).

One well-researched way to curb the flooding overhead is by clustering. Passive clustering is a recent proposal for on-demand creation and maintenance of the clustered structure, making it very attractive for WSNs and directed diffusion in particular.

The contribution of this paper is the investigation of this combination: Is it feasible to execute directed diffusion on top of a sensor network where the topology is implicitly constructed by passive clustering?

A simulation-based comparison between plain directed diffusion and one based on passive clustering shows that, depending on the scenario, passive clustering can significantly reduce the required energy while maintaining and even improving the delay and the delivery rate. This study also provides insights into the behavior of directed diffusion with respect to its long-term periodic behavior, contributing to a better understanding of this novel class of communication protocols.

1 Introduction

Wireless ad hoc networks and wireless sensor networks (WSNs) are similar in many respects: the network has to be self-configurable, scalable, and energy-efficient. The most important difference is the communication model: ad hoc networks typically are built to support standard communication interactions between nodes, where WSNs are centered around data that can be observed from the environment. Owing to the high redundancy of nodes in WSNs, any

---

individual node is not particularly important—it might not even have a unique address—rather, the data is. One appropriate programming model for such a network is publish/subscribe, with data-centric routing as the supporting communication layer.

Directed diffusion [1] is a prominent instance of data-centric routing where the selection of routes is based on application level data. By using purely local interactions and in-network processing it is well tailored to the specific needs of WSNs.

In its operation, directed diffusion relies heavily on performing network-wide broadcasts. Consequently, the overall performance of the protocol can be strongly influenced by the efficiency of this elementary operation.

One of the most straightforward ways to realize a network-wide broadcast is to use the simple flooding algorithm where every node in the network forwards each new message to all of its neighbors. But this approach is rather inefficient in wireless networks where the significant overlap between the neighborhoods of two nodes in immediate vicinity can lead to a large number of unnecessary rebroadcasts. This additionally results in increased channel contention and waste of bandwidth that take further toll on the scarce energy resources of the nodes.

The problem of identifying a set of re-forwarding nodes that still guarantees the distribution of the messages to all of the nodes in the network is very well researched. The proposed techniques vary from probability-based approaches to area- and neighborhood-knowledge methods [2].

The neighborhood-knowledge based group mainly comprises the distributed algorithms for efficient approximation of the minimum connected dominating set of nodes. They are based on the partitioning of the network into logical substructures called *clusters* that confine and mask local interactions from the rest of the network. In the process of defining these substructures (*clustering*) a single node (*clusterhead)* can be selected from every cluster to exert local control and coordination. The interfacing between the different clusters is then realized only by dedicated *gateway* nodes. The selection of the clusterheads and the gateways is such as to guarantee the connectivity of the resulting, reduced topology.

Clustering reduces the flooding overhead by limiting the re-forwarding of the messages to this reduced topology. Yet, for the approach to be overall effective, the cost of building and maintaining the clustered structure has to be lower then the energy savings from reducing the number of sent and received messages.

This requirement is not easily met if clustering is performed for the sole reason of limiting the flooding overhead, especially if one has in mind the pro-active and resource intensive nature of the majority of clustering algorithms. To overcome this problem, Kwon and Gerla [3] proposed the Passive Clustering algorithm (PC) for on-demand creation and maintenance of the clustered substrate.

In this paper, we study if and how directed diffusion can benefit from a passive-clustering-based network topology to reduce its flooding overhead. To do so, we first summarize in Sect. 2 the protocol mechanisms of passive clustering and directed diffusion and discuss our combination of these two mechanisms.

Section 3 then describes the investigation methodology we used to compare original directed diffusion with directed diffusion over passive clustering: the choice of metrics, simulators, and scenarios as well as some implementation details. Section 4 presents the simulation results, and Sect. 5 puts our approach and these results into context with related work. Section 6 concludes the paper.

2   Protocol Description

2.1   Passive Clustering

Passive clustering has several unique properties that increase its viability as a flooding overhead control mechanism for on-demand wireless networking protocols. In the following, we briefly summarize these characteristics.

Unlike in "classical" clustering algorithms, the formation of clusters here is dynamic and is initiated by the first data message to be flooded. In this way, the potentially long initial set-up period is avoided, and the benefits of the reduction of the forwarding set can be felt after a very small number of data message rounds.

Because the main function of the clusters is to optimize the exchange of flooded messages, there is no point in wasting valuable resources to pro-actively maintain such an elaborate structure between floods, when there is no traffic that can make use of it. Consequently, passive clustering refrains from using explicit control messages to support its functionality and all protocol-specific information is piggybacked on the exchanged data messages. This approach joins the frequency of exchanged messages with the quality of the clustered substrate resulting in a graceful trade-off between the "freshness" of the clustered substrate and the introduced overhead for its maintenance.

An additional trade-off is also made over the issue of selection of a clusterhead between several potential candidates. Passive clustering does not have the benefit of exchanging specific control messages to optimally resolve this conflict, as it is usually done in other clustering schemes. It introduces a simple novel rule, called *first declaration wins*. Under this rule, the first aspirer to become a clusterhead is immediately and without further neighborhood checks declared as such and allowed to dominate the radio coverage area. This may at first sound suboptimal, but is more then compensated for by the decreased latency, the prevention of "chain re-clusterings" and the natural correlation that emerges between the traffic flow patterns and the resulting clustered topology.

The effectives of any clustering algorithm as a tool for reducing the number of redundant flood messages directly depends on its ability to select the minimal number of gateways while still maintaining the connected property of the topology. In order to avoid the large overhead of collecting full two-hop neighborhood information that is required for optimal solution of this clustering subproblem, passive clustering resorts to a counter-based gateway and distributed gateway selection heuristic. Under this heuristic, the probability of a node that belongs to two or more clusters to become a gateway is directly proportional to the number

of clusterheads in its radio range and inversely proportional to the number of gateways already present in the same area. By controlling the coefficients of proportionality, one can make trade-offs between the flooding overhead reduction and the level of connectivness in the reduced topology.

As demonstrated by the authors in [4], the combination of these design decisions results in a lightweight and flexible algorithm that can significantly lower the overhead associated with flooding in wireless networks.

2.2   Directed Diffusion

Directed diffusion is in great detail described in [1]. Here, only an overview is given to enable the reader to understand the basic mechanisms as used in our performance evaluation, where passive clustering changes it, and the influence of some protocol parameters.

Directed diffusion is a data-centric routing protocol. In a Wireless Sensor Network (WSN) the data that the network can provide is interesting, not specific nodes. In order to receive data, an interested node (a "sink") floods the network with an interest message. This message contains a detailed description of the event it is interested in (e.g. cars in a certain geographical region). When this interest is received by a node, it sets up a gradient to the neighbor from which it heard the interest. If it hears the same interest form several neighbors, gradients will be set up for each one of them. This focus on neighbors is a specific feature of directed diffusion, which allows the protocol to scale with the number of nodes in the network.

When a gradient has been set up and the node can potentially contribute to this interest (e.g. it can sense data and is in the requested region), the node starts to sense its environment for matching events. When it observes a matching event, it becomes a source and sends the data to all neighbors to which it has gradients. This data is flagged as "exploratory". Exploratory data is forwarded along gradients towards the sink and eventually reaches it; intermediate nodes follow essentially the same rules regarding gradients as sources and sinks. The sink then sends a reinforcement message to only one of its neighbors, namely the one where it heard the exploratory data first. When the source receives a reinforcement, it starts to send data more often, for instance every 2 s. At the same time, it continues to send exploratory data periodically (e.g. every 60 s) to all neighbors to which it has gradients. The gradients are stored only for a limited amount of time (e.g. 125 s), unless a new interest is received. The interests are also sent periodically (e.g. every 20 s).

Directed diffusion maintains a data cache in order to limit flooding. Messages are repeated only once. This (together with some additional protocol messages not described here) ensures that directed diffusion often chooses a single, a "reinforced" path with empirically low delay between sources and sinks.

The periodic flooding of interests as well as the sending of exploratory data (which can easily amount to flooding, as directed diffusion keeps more than one neighbor per received interest) is important to maintain or repair the paths between sources and sinks.
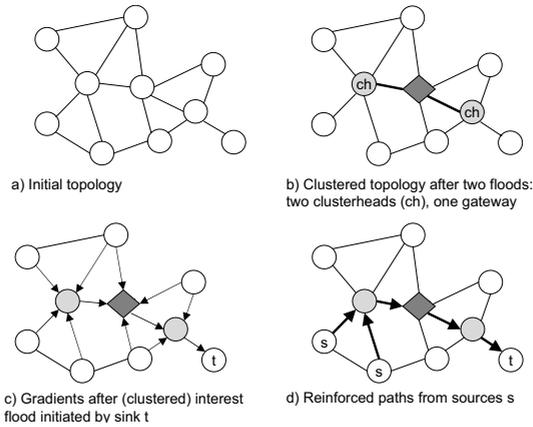
a) Initial topology

b) Clustered topology after two floods:
two clusterheads (ch), one gateway

c) Gradients after (clustered) interest
flood initiated by sink t

d) Reinforced paths from sources s

Fig. 1. Establishing the clustered diffusion structure (gray circles indicate cluster-heads, diamonds indicate gateways)

2.3   Directed Diffusion and Passive Clustering

From the description in Sect. 2.2 it is evident that the propagation of interests and exploratory data are one of the most resource-demanding operations in directed diffusion. Depending on the actual network topology, their implementation using simple flooding rules can lead to significant and unnecessary overhead that reduces the operational lifetime of the sensor network.

We believe that passive clustering is an appropriate mechanism to increase the efficiency of these crucial diffusion phases and by that the efficiency of the whole protocol. The protocols share some common properties:

1. They are on-demand mechanisms that maintain the operation only while there is application traffic in need of their services
2. They rely on purely local information for performing their functions

As a result, their combination is perfectly adapted to the operating conditions that are common in WSNs.

For the initial exploration, we decided to limit the interaction between them just to a single point in the operation of the original directed diffusion:

Ordinary (non-clusterhead, non-gateway) nodes do not forward the interest and exploratory data messages that they receive.

To explain the effects of this interaction, we demonstrate in Fig. 1 the establishment of the directed diffusion routing structures when used in combination with passive clustering.

Let the operation start with the topology depicted in Fig. 1(a). In passive clustering, nodes that are first able to piggyback information onto an outgoing

message (see Sec. 2.4) will declare themselves as clusterheads (assuming that they have not heard of other clusterheads before). Nodes that have heard of two clusterheads but of no other gateway assigned to these same clusterheads will become a gateway, resulting in the clustered topology shown in Fig. 1(b). Assuming this constellation, the next diffusion *interest* flood will establish a sparse gradient structure shown in Fig. 1(c). The reinforced data paths are then shown in Fig. 1(d).

Based on this description, we can make a couple of observations:

1. The number of redundant transmissions during the broadcasting tasks is significantly reduced
2. The size of all directed diffusion messages that are broadcasted is increased by the size of the passive clustering header
3. Passive clustering reduces the level of connectivity available to directed diffusion

This means that the overall effectiveness of our approach depends on the answers to the following crucial questions:

– Does the combination of the reduced number of messages with the increased message size result in net gain in energy efficiency?
– Does the reduced level of connectivity result in degradation of the data distribution capability?
– Does the reduced level of connectivity result in increase of the latency as sometimes less optimal paths have to be followed?

We shall answer these questions in Sect. 4. But first, a few remarks on the implementation of directed diffusion with passive clustering are in order.

2.4   Implementation

The interfacing between our modules and the diffusion core was realized using the provided Filter API [5]. The node's passive clustering component is declaring two filters to the diffusion core, which bracket the diffusion gradient filter: One filter with high priority to intercept messages before gradient processing (*pre-filter*) and a second filter with lower priority for messages from gradient to the network (*post-filter*). The interaction of passive clustering with the diffusion routing core is shown in Fig. 2.

Due to the reactive nature of the protocol, all state changes happen upon interception of a diffusion message. Passive clustering attaches a header to outbound messages and inspects it in incoming messages.

Here, the notion of inbound and outbound messages in diffusion is somewhat different from the usual network stack view: Messages arrive at the diffusion routing core (point (1) in Fig. 2), both from a local application or from the network. After passing the filters (8), the message destination may again be either the network or a local application. All messages pass the filter in the same
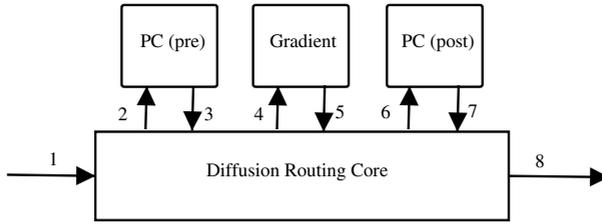
Fig. 2. Integration of Passive Clustering and Diffusion

order (according to the given numbering in Fig. 2), independent of their source. In a network stack view, passive clustering would be located below the gradient layer.

If a message has been received from the network, the *pre-filter* processes the message header and changes the node's internal state (Tab.1) according to the newly available information about a neighbor. This includes entering certain "candidate" states, which denote that the new information is relevant for the node to change its external state on the next outbound message. Pre-filter does not process any messages coming from the local application. After inspecting a message for clustering-relevant information, the message is then forwarded to the diffusion gradient filter.

The final state-change to an externally visible state like clusterhead *(CLUS-TER_HEAD)* or gateway *(GW_NODE, DIST_GW)* is performed on the next outbound message addressed to the network that is intercepted by the *post-filter*.

Table 1. Passive clustering FSM states

Internal States	External States
CH_READY	INITIAL_NODE
GW_READY	CLUSTER_HEAD
	GW_NODE
	DIST_GW
	ORDINARY_NODE

A passive clustering header (Fig. 3) is piggybacked onto this message *only* if the external state of the node has changed or if it has not announced its current state for more than a certain parametrized number of seconds. This keeps passive clustering overhead to a minimum, as it does not grow with the number of messages but is bounded by $O(\#nodes)$.

The last two fields (CH1_ID and CH2_ID) are present only in the messages issued by gateway nodes and represent the IDs of the cluster heads they connect.

Fig. 3. Structure of the piggybacked passive clustering header

Including the 4 byte overhead caused by attaching the header as an attribute to
the diffusion message, the fields sum up to a total of 17 bytes in the worst case.

The essential task of discarding the unnecessarily forwarded messages is also
performed by the *Post-Filter*: both *Interest* and *Exploratory Data* messages are
discarded if the node is in (or has changed to) *Ordinary Node* state. This happens
unless, of course, messages originate from a local application that is resident on
this node or are addressed to it.

3 Methodology

Designing performance evaluation experiments for wireless sensor networks is
faced with a number of practical and conceptual difficulties. This section sum-
marizes our main choices for the simulation setup.

3.1 Choice of Protocol Versions

We simulated the original directed diffusion and our passive clustering extension
using the ns-2 simulator [6], version 2.26. Directed diffusion is implemented for
this simulator in two versions. The first version is a simplified implementation of
directed diffusion that was used by Intanagonwiwat et al. [1] for the evaluation of
the protocol. Additionally, directed diffusion 3 is available; it is a complete pro-
tocol implementation, not a simplified version. The implementation is described
in [1] and allows a more realistic evaluation of the protocol.

The protocol versions have some major differences. First of all, the simplified
version uses somewhat arbitrary lengths for different message types. The interest
messages, for example, are half the size (32 bytes) of data messages (64 bytes).
For the protocol implementation, the difference is not as large, since the interest
contains a fairly complete description of the event the sink is interested in. In
our case, the interest message is 88 bytes long.

As this newer version better reflects reality, we decided to base our simula-
tions on it, although this makes comparison with the published results difficult.
We used the "ping" application which adds a sequence number and a time-stamp
to the payload of a data message, resulting in messages that are 116 bytes long.

3.2 Load Model

A second major difference are the timer settings of these protocols. The sim-
plified version floods an interest message every 5 s and an exploratory message

every 50 s. On a reinforced gradient, the sources send data every 0.5 s. The standard settings of these parameters for the protocol implementation are different. Interests are sent every 25 s, exploratory data every 60 s and the sources send data every 5 s. Here, only five data messages per interest flood are received by a sink. In this case, the protocol overhead for directed diffusion is very large, making it questionable whether it pays off at all compared to plain flooding.

In our simulation we changed the ping application to send data every 2 s and interest messages every 20 s, which results in 10 data messages per interest flood, the same as in the simplified version. As in [1] we regard every such message as a "new distinct event" and we do not use duplicate suppression. Duplicate suppression, or aggregation as it is called in [1], allows nodes in the network to drop messages if it is an event that they have already forwarded. However, duplicate suppression is difficult in simulations, as there is no commonly used model that produces the "right" amount of duplicates.

3.3   Energy Model

In a WSN, energy is a scarce resource and should be saved. Hence, energy consumption per data packet (or per distinct event) is our main metric. The problem is how to measure this energy. Clearly, every sending of a message consumes power and the number of sent messages should be reduced as far as possible. In addition, every message that was received by a node when it is not the final destination is overhead and the power used for receiving messages should be reduced, too.

A further problem is the idle power, the power a node consumes when it is listening for transmissions. This is tightly connected with the used MAC; in the case considered here, the IEEE 802.11 implementation of ns-2 as used in the original directed diffusion investigations. For such a CSMA MAC the node has to listen to the channel permanently. Hence, for a CSMA MAC, the idle power is comparable to the receive power. Although this would be technically correct, the idle power could mask the effects at the network layer. With long application related inactivity periods it would dominate the energy budget making the comparison between the two protocols difficult. Since we wanted to concentrate on the relative performance of the protocols, we used a zero idle power in our simulations.

For comparability we used the same energy model as in [1] that is the PCM-CIA WLAN card model in ns-2. This card consumes 0.660 W when sending and 0.395 W when receiving.

3.4   Metrics

The characteristics of the ping application result in two performance metrics. One metric is the delivery ratio, which compares the number of messages arriving at the sinks with the number of messages sent by the sources. Another metric is the delay—how much time does a message need to travel from the source to the sink.
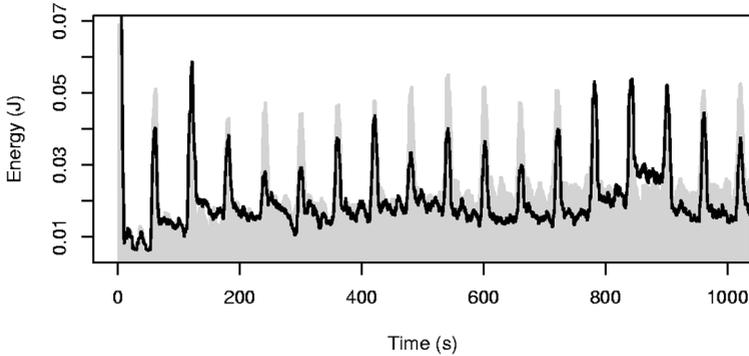
Fig. 4. Time series of the energy per distinct event and sink, showing the initial transient. Plain directed diffusion – gray in the background, diffusion with passive clustering – black in the foreground.

Every time a new distinct event is seen, the energy consumed since the last distinct event is computed for every node in the network. From these values the average is computed and reported—this constitutes our third performance metric, average dissipated energy (per new distinct event). While the energy consumption is reported only once per distinct event, the delay and delivery ratio are reported by each sink separately, as this is more natural for these metrics.

Figure 4 shows an example for the energy consumption per distinct event and sink as a function of time (to display this figure properly, the actual data has been slightly smoothed by a moving average with a small window size). A close inspection of the time series of the energy consumption revealed a long initial transient phase, which lasts for about 300 s in the example shown in Fig. 4. This transient phase is in part due to the staggered subscriptions and publications performed by all nodes. To correctly handle the initial transient, we discarded the observation values produced during the first 600 s from every simulation run; to compute the average energy consumption per new distinct event, we took into account only the values from the remaining 1000 s of simulated time. The passive clustering structure is refreshed every 45 s, so discarding of the first 600 s does not skew the results in favor of the passive clustering approach.

3.5   Scenarios

Every node has a fixed transmission power resulting in a 40 m transmission range. The sources and sinks were spread uniformly over the entire area; the size of the area varies between simulations.

We used two different types of source/sink setups.

Five sources and five sinks. With five sources and sinks each distributed as described above, either the area or the average density of the network are

kept constant: an increasing number of nodes is placed either in an area of fixed size (160 m by 160 m) or on a growing area.

Variable number of sources/sinks. Using a 160 m by 160 m area with 100 nodes, we used either a single sink with a variable number of sources or a single source with a variable number of sinks.

For each parameter setting, we randomly generated 10 scenarios with different placements of nodes. For each scenario, we computed (after removing the initial transient) the means for all three metrics. From these ten (stochastically independent) means we computed the means over all scenarios and their 95% confidence intervals for all metrics.

4   Results and Discussion

4.1   Effects of the Network Topology

We start the presentation of our results with the five sources/five sinks case in networks of either fixed area or fixed density. Figure 5 depicts the average dissipated energy per unique event for the original directed diffusion protocol (DD) and for the directed diffusion with passive clustering (PCDD) under three different network sizes. For each different network size, the simulation area is scaled so that the average node degree in the network remains constant at around 10. This enables us to examine the scaling properties of the protocols as a function of the number of the nodes in the network in isolation. In all of the topology experiments the load of the network is generated by five sources sending unique events every two seconds.
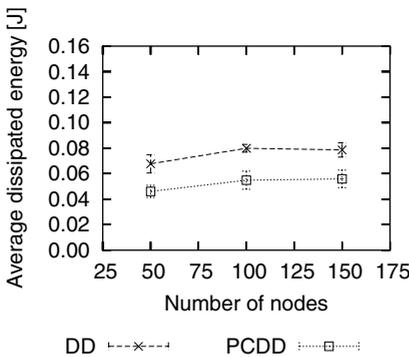

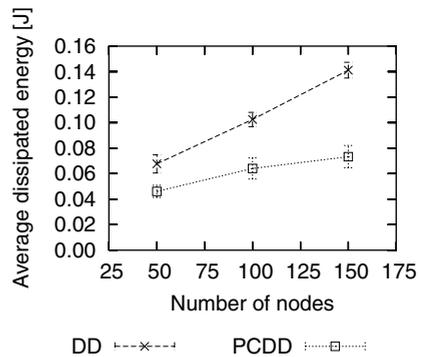**Fig. 5.** Energy efficiency – fixed density scenario

Fig. 6. Energy efficiency – fixed area scenario

The results clearly show the effectiveness of passive clustering in controlling the unnecessary re-forwarding of messages in directed diffusion. While both protocols show slow degradation of their performance with the increase in number

of nodes, the relative savings of the passive-clustering augmented directed diffusion over the original protocol is evident even for the 50 nodes scenario and is maintained as the size of the network increases. This is a confirmation to our assumption that the reduced number of flooded messages results in a net gain in the energy metric, despite the increased size of the remaining messages.

Importantly, this gain was not achieved at the cost of either the delivery ratio (remaining above 0.95 for both protocols) or the delay (remaining below 0.6 s for both protocols). The corresponding figures are omitted due to space constraints but it is obvious that the resulting connectivity after the passive filtering allowed these metrics to remain close to their optimal values for the whole range of simulated network sizes.

As described in Sect. 2, both directed diffusion and passive clustering are based on pure local interaction. Many of the elementary operations of which they are comprised operate over the set of neighboring nodes that are in radio range. As a result, their complexity, and by that, their performance is strongly determined by the size of that set, i.e., on the average node degree in the network. Again, we have examined our performance metrics for three different network sizes as previously, but now keeping the simulation area constant to the initial 160 m by 160 m. This gives us an insight into the performance of the evaluated protocols under an average number of around 10, 20, and 30 neighbors.

In Fig. 6 we can see the average dissipated energy per unique event for an increasing number of nodes in this fixed area. The original directed diffusion shows rather linear increase in the required energy with the increase of the number of neighbors. This was to be expected, as the increased number of neighbors results in higher number of flooded interest messages and an increase in the number of unicasted exploratory data messages. Passive-clustering-supported directed diffusion, on the other hand, behaves much better in dense networks. Like in the fixed degree case, the increased overhead due to the PC headers is more then compensated by the reduction of the costly send and receive operations. The denser the network, the larger the advantage over pure directed diffusion as more redundant paths can be suppressed by passive clustering.

Moreover, these advantages in energy efficiency are not paid for in a reduced delivery rate. As Fig. 7 shows, the contrary is true: the delivery rate actually is considerably better with passive clustering. Compared with the fixed degree scenario, the delivery ratio of plane directed diffusion experiences significant degradation as the size of the neighborhood increases. By reducing the superfluous message exchanges, passive clustering leads to a decrease in the collision rate that results in a very good delivery ratio even for the highest density case.

Similarly, Fig. 8 shows that also the delay is much better with passive clustering, despite the potential problem of the longer paths. The reduced load in the network more than compensates for this potentially negative aspect. Without this reduction in the amount of traffic in the network, plain directed diffusion faces extensive delays in the higher degree scenarios.

Overall, we can conclude that passive clustering augmented directed diffusion scales much better with respect to the neighborhood size, maintaining a satisfactory performance in the progressively harsher network environment.
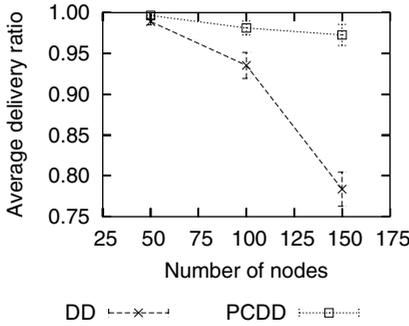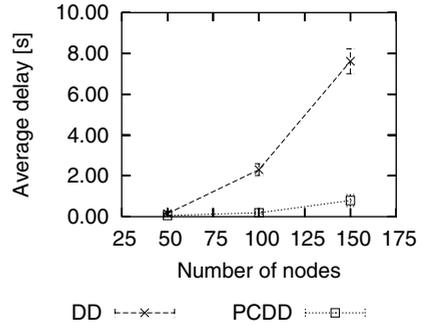

Fig. 7. Delivery ratio – fixed area scenario

Fig. 8. Delay – fixed area scenario

4.2   Effects of the Traffic Pattern

What happens when the number of sinks or sources is varied, as described in the second type of scenarios in Sect. 3.5?
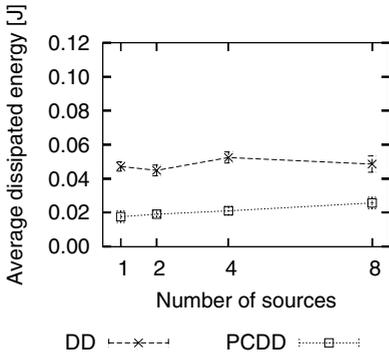

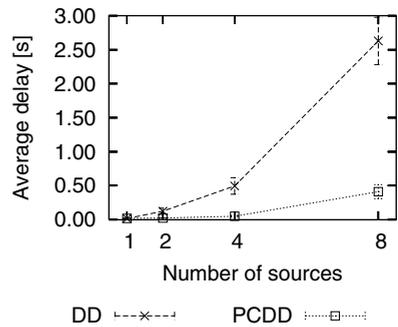**Fig. 9.** Energy efficiency – single sink scenario

Fig. 10. Delay – single sink scenario

As one could expect, the relative gain of the passive clustering enhancements compared to plain directed diffusion increases with the number of sources: More sources issue more exploratory data floods, which in turn benefit from passive clustering. Our energy metric shown on Fig. 9 somewhat masks these effects

because the average dissipated energy *per event* tends to drop as the number of generated events due to the multiple sources increases. The effects of the increased load and the performance gains from the passive clustering are much more visible on the delay plot shown on Fig. 10.

A similar pattern becomes apparent when the number of sinks is varied while having only one source. Here the amount of "sensed" data is not increased, but each data message needs to be reported to more locations in the network. This, combined with the increase in the "interests" traffic, leads to a rise in the energy consumption as shown in Fig. 11. The delay (Fig. 12) also ncreases with the number of sinks, but not as dramatically as with the number of sources.
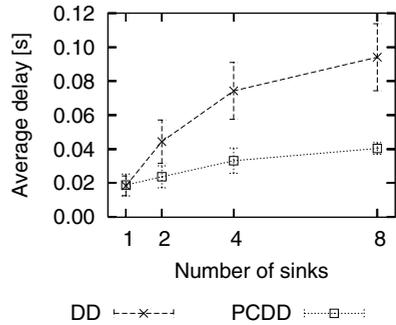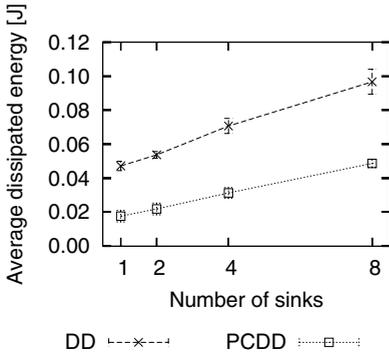

Fig. 11. Energy efficiency – single source scenario

Fig. 12. Delay – single source scenario

5   Related Work

The optimization of the data distribution process in WSNs has been the focus of interest in a couple of recently published papers; the most relevant ones are discussed here.

In [7], the authors concentrate on the problem of efficient distribution of queries that do not excite large amount of data in response. For these *one-shot* queries it is usually more efficient to directly send data along an even non-optimal path, foregoing optimization overhead. They present the *rumor routing* algorithm where the query is sent on a random walk until it intersects an already established path to the event. In contrast to this active networking approach, the passive clustering and directed diffusion combination is better suited for the more long-lived data gathering applications in WSNs. Under these conditions, it is worthwhile to explore the network topology in order to find the optimal paths between the sinks and the sources, as they are going to be used by a (likely) large amount of data during the lifetime of the application.

When the nodes have information about their own and their neighbors' geographic location, geo-routing and geographical overlays become another potential tool for limiting the overhead. Ye et al. [8] present the *Two-tier data dissemination* approach for efficient distribution in large sensor networks with multiple sources and multiple *mobile* sinks. They create grid distribution overlays over which the sources pro-actively announce the availability of new data. The flooding of the queries, initiated by the moving sinks, is then confined to a single cell of this overlay and does not create a network wide restructuring of the routing structure. Our approach does not assume the availability of location information in the network. Instead, we are indirectly sensing the density of the nodes in our neighborhood and try to dynamically adjust the forwarding topology such that the redundant rebroadcasts are eliminated as much as possible.

The authors of [9] describe the *Cluster-based Energy Conservation* protocol for topology control in WSNs, which is closest to our work. Their clustering algorithm also results in the formation of 2-hop clusters. Instead of the *first declaration wins* rule that is used in passive clustering, CEC selects the clusterheads based on the estimated lifetime of the nodes. The gateway selection heuristics is similar as it is influenced by the number of clusterheads and already promoted gateways in the vicinity. But major differences also exist. The CEC is positioned as a general-purpose topology and scheduling mechanism for WSNs. Our approach, on the other hand, is specifically focused on limiting redundant retransmissions during the flooding phases of directed diffusion. One of the major benefits of the passive clustering is the fact that it does not require an active maintenance of the clustering structure, something that is crucial for the operation of CEC. We believe that its properties are a better match to the problem at hand then any other "active" clustering approach.

A number of other cluster-based routing protocols have also been proposed, e.g., LEACH [10]. These approaches share the disadvantages of active clustering or are otherwise not directly applicable to the support of directed diffusion.

6   Conclusion

Several distinct conclusions can be drawn from the results in this paper. First of all, we showed that flooding consumes a large part of the energy in directed diffusion. When flooding of interests and exploratory messages is limited using passive clustering, this will result in large energy savings. But not only the energy savings are important. The decreased load also leads to a better delivery ratio and lower delay, as fewer messages are lost due to collisions.

While the passive clustering in the form presented here is very beneficial, it can introduce hot spots in the network, namely the gateways between two clusterheads and the clusterheads themselves. Their energy expenditure is much higher than those of ordinary nodes. How to deal with this asymmetry between nodes on top of an intentionally passive clustering approach is an open problem. We intend to compare our approach with active clustering approaches under this perspective.

Further extensions could make use of the clusterheads as natural points to aggregate the data. This aggregated data is more important than not aggregated data, which could make it beneficial to transmit it redundantly using multiple gateways instead of using a reliable MAC.

In this paper, we have consciously decided to focus on network-level improvements of directed diffusion and ignored any possible gains from a modification of the MAC layer. For example, the decision to unicast exploratory data to all neighbors, although logical from the routing protocol point of view, harms the performance. We have performed some preliminary experiments which used MAC broadcasts instead of MAC unicasts for these packets. These experiments show a substantial improvement over the unicast case.

This simple modification, however, does not remove all remaining problems, as some of the difficulties stem from the traffic patterns. For the highly correlated and periodic traffic produced by directed diffusion, the IEEE 802.11 MAC is unsuitable. Thus, the interaction and the integration of directed diffusion with the underlying MAC merits further investigation.

References

1. Intanagonwiwat, C., Govindan, R., Estrin, D., Heidemann, J., Silva, F.: Directed diffusion for wireless sensor networking. IEEE/ACM Transactions on Networking (TON) **11** (2003) 2–16
2. Williams, B., Camp, T.: Comparison of broadcasting techniques for mobile ad hoc networks. In: Proceedings of the third ACM international symposium on Mobile ad hoc networking & computing, ACM Press (2002) 194–205
3. Kwon, T.J., Gerla, M.: Efficient flooding with passive clustering (PC) in ad hoc networks. ACM SIGCOMM Computer Communication Review **32** (2002) 44–56
4. Yi, Y., Gerla, M.: Scalable AODV with efficient flooding based on on-demand clustering. ACM SIGMOBILE Mobile Computing and Communications Review **6** (2002) 98–99
5. Silva, F., Heidemann, J., Govindan, R.: Network routing application programmer's interface (API) and walk through 9.0.1. Technical report, USC/ISI (2002)
6. UCB/LBNL/VINT: Network simulator - ns-2. (http://www.isi.edu/nsnam/ns/)
7. Braginsky, D., Estrin, D.: Rumor routing algorithm for sensor networks. In: Proceedings of the first ACM international workshop on Wireless sensor networks and applications, ACM Press (2002) 22–31
8. Ye, F., Luo, H., Cheng, J., Lu, S., Zhang, L.: A two-tier data dissemination model for large-scale wireless sensor networks. In: Proceedings of the eighth annual international conference on Mobile computing and networking, ACM Press (2002) 148–159
9. Xu, Y., Bien, S., Mori, Y., Heidemann, J., Estrin, D.: Topology control protocols to conserve energy in wireless ad hoc networks. Technical Report 6, University of California, Los Angeles, Center for Embedded Networked Computing (2003) submitted for publication.
10. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: Proc. of the 33rd Hawaii Intl. Conf. on System Sciences. (2000)