

# Energy Efficient Network Structure for Synchronous Preamble Sampling in Wireless Sensor Networks

Fabrice Theoleyre\* Abdelmalik Bachir§, Nesrine Chakchouk† Andrzej Duda\* Kin K. Leung§

\* Grenoble Informatics Laboratory, Grenoble, France

† EECS, Oregon State University, USA

§ Imperial College, London, UK

**Abstract**—We propose a new energy efficient network structure for maintaining synchronization in access methods based on Synchronous Preamble Sampling. Our scheme limits the number of synchronization messages and increases network capacity through the use of multiple non-interfering *virtual channels*. It consists in constructing independent clusters based on the Weakly Connected Dominating Set (WCDS) so that clusters can use different virtual channels and only need to maintain internal synchronization, while still offering global connectivity. We define a distributed and self-stabilizing algorithm for constructing and maintaining the clusters. Our simulation results show that the proposed scheme has comparable energy consumption as Scheduled Channel Polling, but results in better network capacity. Moreover, it achieves better energy savings and network capacity than recently proposed Crankshaft access method.

## I. INTRODUCTION

Most of the research on wireless sensor networks has focused on energy saving to increase node lifetimes. As radio communications are responsible for most energy consumption, many proposals aim at optimizing the MAC layer and its use of radio transmissions. Preamble Sampling [1], also referred to as Low Power Listening [2], is a key technique for reducing *idle listening* in sensor networks. Idle listening is undesirable, because nodes consume energy while listening to an idle channel [3]. It mainly occurs in contention-based access methods in which nodes listen to the channel for long periods of time, because they do not know the exact instant at which other nodes may send a frame. In preamble sampling, a node sends a *preamble* before each data frame, while nodes wake up each *check interval* to sample the channel for the preamble and receive a frame—if the preamble lasts for at least one check interval, the receiver surely wakes up during the preamble transmission. After sensing the preamble, potential receivers must stay awake to receive a data frame.

In Synchronous Preamble Sampling, nodes coordinate their wake-up instants to have a common schedule—they sleep during the same interval to save energy and then wake up to exchange data frames [4]. To maintain a common schedule, each node needs to periodically broadcast a SYNC control frame to make up for clock drifts.

Although this method drastically reduces energy consumption by minimizing idle listening, it presents some shortcomings. First of all, SYNC signaling frames represent significant overhead that increases energy consumption. Moreover, when nodes wake up, they synchronously contend for the radio channel, which results in collisions that waste bandwidth and energy. Finally, the overall network capacity is lower, because

all transmissions happen in the same common wake-up slot, while nodes sleep the rest of the time.

To overcome these shortcomings, we propose to structure the network in a way that limits the number of synchronization messages and increases network capacity through the use of multiple *virtual channels*. Our approach consists in constructing independent clusters that maintain their own synchronization and use their own virtual channels while maintaining global network connectivity. We define a virtual channel as a means for communication in a group of nodes through the use of any combination of time (TDMA), frequency (FDMA), or code (CDMA) multiplexing. For instance, Wavenis products [5] combine different time slots (or schedules) and specific frequency hopping sequences per time slot to define multiple virtual channels. Two nodes that use different virtual channels cannot directly communicate, but can operate in parallel, which increases the overall network capacity.

To reduce the number of exchanged SYNC frames while maintaining the network connected, we structure it according to a Weakly Connected Dominating Set (WCDS) [6]. A WCDS is composed of independent clusters that can use different virtual channels for internal communication, which only requires maintaining synchronization inside one cluster. Besides, all transmissions can use short Preamble Sampling because of the WCDS structure property. Thus, only a subset of nodes needs to transmit SYNC frames instead of all nodes in traditional Synchronous Preamble Sampling.

The contribution of this paper is threefold:

- we propose a new network structure based on a WCDS to limit energy waste for maintaining synchronization while increasing network capacity through the use of multiple non-interfering virtual channels;
- we present a simple distributed self-stabilizing algorithm for a WCDS that leads to two variants: full-WCDS bridged-WCDS;
- we evaluate the performance of Synchronous Preamble Sampling over the WCDS structure with virtual channels and compare it with major state-of-the-art protocols.

## II. RELATED WORK

WiseMAC [7] is among the first implementations of synchronous preamble sampling. Each node maintains its own schedule and piggybacks synchronization information in its `ack` frames. WiseMAC improves network capacity by setting up multiple virtual channels, but long preambles are required to transmit broadcast frames or when synchronization is lost between source and destination nodes.

Scheduled Channel Polling (SCP) [4] has shown that explicit transmissions of SYNC frames do not reduce the overall energy savings, because the benefits of using short preambles lead to more energy savings in the end. SCP maintains a common schedule throughout the network to save broadcast communications the need of using long preambles. With such a design, SCP trades capacity for energy savings, because contention drastically increases when nodes use the same virtual channel (i.e. the same wake-up schedule).

Crankshaft [8] reduces contention by using preamble sampling on top of a TDMA scheme. TDMA slots are receiver-oriented, i.e. each node listens to its own virtual channel for receiving frames. If none is received, it immediately goes back to sleep. Since each node has its own virtual channel, Crankshaft increases network capacity.

Y-MAC [9] uses a similar approach, but multiplies virtual channels when traffic increases. Initially, it assigns slots to each receiver like Crankshaft, but a node can negotiate additional virtual channels for each of its neighboring senders. In this way, Y-MAC adapts to traffic conditions: the network uses more virtual channels when more frames need to be forwarded. Wavenis [5] adopts a similar approach, but adds a combination of slow and fast frequency hopping between physical channels to further increase robustness to interference.

Although Crankshaft, Y-MAC, and Wavenis increase capacity by reducing contention through the use of multiple virtual channels, they suffer from an additional overhead related to the broadcast channel—under these access methods, nodes need to listen to the additional broadcast channel for receiving broadcast communications. Therefore, the energy drained in periodically checking the channel is doubled, which results in an important energy waste especially in low traffic data networks.

### III. ENERGY COST FOR SYNCHRONOUS PREAMBLE SAMPLING

The power drained in Synchronous Preamble Sampling consists of the power consumed during channel sampling and the power spent on sending and receiving SYNC frames. Both of these operations are periodic. Channel sampling is performed every Check Interval  $T_{CI}$  and SYNC frames are transmitted every Synchronization Interval  $T_{SI}$ .

We call  $E_{\text{sampling}}$  the energy drained during a channel sampling operation. Therefore, the mean power drained in channel sampling is:

$$\mathcal{P}_{\text{samp}} = \frac{E_{\text{sampling}}}{T_{CI}} \quad (1)$$

The cost of transmitting a frame (SYNC or another frame) depends on the accuracy of synchronization between the transmitter and the receiver. As synchronization is not perfect, the transmitter does not know the exact wake-up time of the receiver. Therefore, the transmitter sends a short preamble before the data frame to make sure that the receiver finds the channel busy when it wakes up to sample the channel. When the receiver finds that the channel is busy, it keeps listening to it until it receives the data. The length of the short preamble

$T_{\text{preamble}}$  depends on the clock drift  $\theta$  and the periodicity of maintaining synchronization ( $T_{SI}$ ) [7]:

$$T_{\text{preamble}} = 4\theta T_{SI} \quad (2)$$

Note that the length of the preamble does not need to be larger than  $T_{CI}$ , because each node wakes up every  $T_{CI}$  to check for transmissions.

When exchanging SYNC frames, a node may be a *reference* and/or *follower*. A node is said to be a reference, if it transmits its clock to its neighbors and is said to be a follower, if it adjusts its clock to the time received from a reference. Nodes with synchronized clocks can use short preambles for their transmissions thereby avoiding the use of costly long preambles. In this case, we will refer to the link between the two nodes as *synchronized*.

The mean power drained by a reference (resp. a follower) node  $\mathcal{P}_{\text{ref}}$  (resp.  $\mathcal{P}_{\text{foll}}$ ) corresponds to cost of transmitting (resp. receiving) a SYNC frame, which is equal to:

$$\mathcal{P}_{\text{ref}} = \left( \frac{T_{\text{preamble}} + T_{\text{SYNC}}}{T_{SI}} \right) P_{\text{tx}} = \left( 4\theta + \frac{T_{\text{SYNC}}}{T_{SI}} \right) P_{\text{tx}} \quad (3)$$

$$\mathcal{P}_{\text{foll}} = \left( \frac{\frac{T_{\text{preamble}}}{2} + T_{\text{SYNC}}}{T_{SI}} \right) P_{\text{rx}} = \left( 2\theta + \frac{T_{\text{SYNC}}}{T_{SI}} \right) P_{\text{rx}} \quad (4)$$

where  $P_{\text{tx}}$  (resp.  $P_{\text{rx}}$ ) is the power drained in the transmit (resp. receive) mode and  $T_{\text{SYNC}}$  is the transmission time of a SYNC frame. Note that the mean preamble reception time in Eq. 4 is a half of the preamble, because the receiver may wake up at any time between the beginning and the end of the preamble.

### IV. NETWORK STRUCTURE FOR MAINTAINING SYNCHRONIZATION

Our objective is to minimize the cost of maintaining synchronization. To achieve this, we want to reduce the number of reference nodes, but in a way that preserves connectivity over synchronized links between any two nodes in the network. Another objective is to increase the overall network capacity that we can achieve by using multiple virtual channels.

#### A. Preliminaries and Definitions

Limiting the number of reference nodes while maintaining network connectivity relates to the problem of constructing a Weakly Connected Dominating Set (WCDS) so we introduce this notion more formally. We represent a sensor network as an undirected graph  $G = (V, E)$ , where  $V$  is the set of nodes and  $E \subseteq V^2$  the set of edges. An edge  $(u, v)$  belongs to  $E$  if  $u$  can communicate with  $v$ . The set of neighbors of a node  $u$  is  $N(u)$ . Formally, the WCDS we want to find satisfies the following conditions:

$$\forall d \notin \text{WCDS}, \exists d' \in \text{WCDS} | d' \in N(d) \quad (5)$$

$$G(V, E') \text{ connected} | E' \supseteq E' = \{(u, v)\}, u \in \text{WCDS}, v \in V \quad (6)$$

The nodes belonging to the WCDS set are called *dominators* and the other ones *dominatees*. From the synchronization

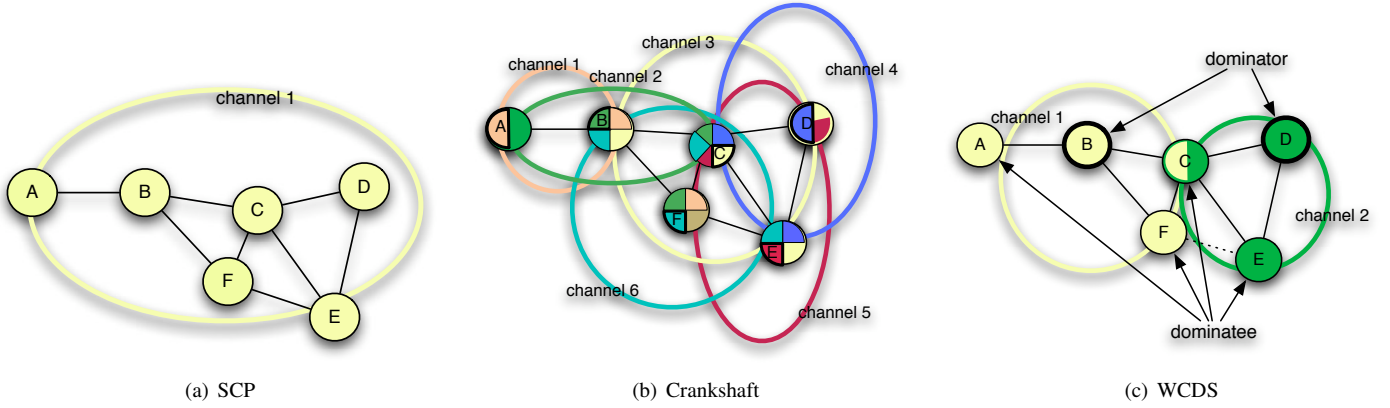


Fig. 1. Synchronization Maintenance.

point of view, dominators are references while dominatees are followers. All links between dominators and dominatees are synchronized—the nodes can use short preambles: a dominatee uses for its receptions/transmissions the virtual channel established by the dominator. Moreover, a path that uses only the links between dominators and dominatees exists in the network between any pair of nodes; it is one of the main property of a WCDS. Thus, the network can exclusively use synchronized links to forward traffic.

A dominator and the set of its neighboring dominatees form a *cluster*. A dominator defines the virtual channel used in its cluster and maintains it with periodical SYNC frames. The dominatees just use this virtual channel for their transmissions in the cluster. Nodes belonging to more than one cluster may act as bridges to forward traffic from one cluster to another.

### B. Motivating Example

Let us consider the cost of running Synchronous Preamble Sampling under different access methods. Consider the network depicted in Figure 1 and let us focus on the energy drained by node *C* as an example.

In SCP, each node transmits a SYNC frame to propagate the same schedule throughout the entire network. As a consequence, each node receives as many SYNC frames as the number of its neighbors. In addition, each node needs to sample the channel to check for incoming frames. Thus, Node *C* consumes a mean power equal to  $\mathcal{P}_{\text{ref}} + \mathcal{P}_{\text{samp}} + 4\mathcal{P}_{\text{foll}}$ .

In Crankshaft, each node consumes similar energy as in SCP to maintain synchronization and to check for incoming unicast traffic. As broadcast communications are performed on another virtual channel (slot), each node needs to sample two channels. Thus, Node *C* consumes a mean power equal to  $\mathcal{P}_{\text{ref}} + 2\mathcal{P}_{\text{samp}} + 4\mathcal{P}_{\text{foll}}$ .

In our WCDS-based structure, Nodes *B* and *D* are dominators and act as references. Node *C* acts as a bridge between the clusters of *B* and *D*: it listens to both virtual channels (channel 1 maintained by *B* and channel 2 maintained by *D*). Node *C* does not transmit SYNC frames, so it consumes a mean power equal to  $2 * (\mathcal{P}_{\text{foll}} + \mathcal{P}_{\text{samp}})$ . Compared to the other schemes, the power spent on maintaining synchronization is reduced.

### C. Distributed Construction of WCDS

Several centralized algorithms exist for constructing a WCDS [10], [11]. We provide below (Algorithm 1) a distributed version that takes into account node failures and frame losses.

---

#### Algorithm 1 Distributed WCDS construction

---

```

1: myRole ← idle
2: loop
3:   id,role,seqnum ← extractInfo(lastHello)
4:   #Age information
5:   if seqnum > largestSeqnum then
6:     largestSeqnum ← seqnum
7:   end if
8:   #Determine role
9:   if myRole = idle then
10:    if role = dominator then
11:      myRole ← dominator
12:      myOriginator ← id
13:    else if role = dominatee then
14:      startTimer(myDegree, myId)
15:      myOriginator ← id
16:    end if
17:  end if
18:  #Maintain connectivity
19:  if (Timeout(myOriginator)) or ((myOriginator = id) and
  (role = idle or seqnum < largestSeqnum - Δseq)) then
20:    myRole ← idle
21:  end if
22:  #Become dominator
23:  if elapsedTimer() and myRole = idle then
24:    myRole ← dominator
25:  end if
26: end loop

```

---

We define three node states: *idle* (initial state), *dominator* (reference) and *dominatee* (follower). At the beginning, all nodes are in the idle state. We assume that the sink node becomes the first dominator, then each node in the idle state applies the following rules:

- If it receives a `hello` message from a dominator, it becomes a dominee.
- If it receives a `hello` message from a dominee, it starts a timer inversely proportional to its weight (combination of the degree and an id). If the timer expires and the node did not receive a `hello` message from a dominator, it becomes a dominator.

Timers used by nodes to determine their roles make the algorithm robust to frame losses. Even if a `hello` message is lost, other nodes can quickly determine their roles.

When a node changes its state, it saves the id of the originator, which is the node from which it received the corresponding `hello` message. In this way, we maintain a tree rooted at the sink. It can be easily verified that nodes alternate their roles along the tree. Thus, we construct a valid WCDS: any node in the network can be joined through a path containing only links of the form (dominator,dominee) or (dominee,dominator).

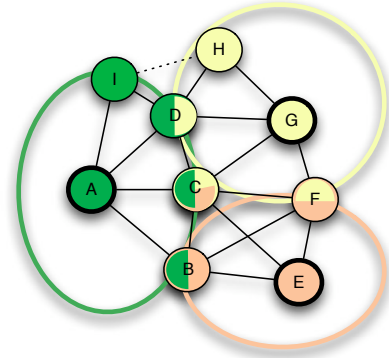
The sink increments the sequence number in each of its `hellos`. A node includes the largest sequence number received from its originator (i.e. the node that triggers its state change) in its own `hellos`. If the sequence number remains unchanged for a long time, this means that the node is probably disconnected from the sink. Thus, the node reinitializes its state, i.e. it becomes idle. The node will choose a new state only if the sequence number of a `hello` message is strictly superior to the last received sequence number— in this way, we avoid loops in the tree. A node that loses its originator proceeds in the same way.

#### D. Variants of the WCDS Structure

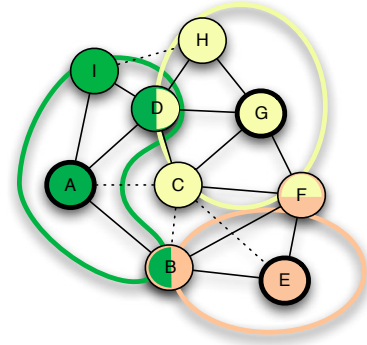
As each WCDS cluster uses its own virtual channel, bridge nodes need to maintain connectivity between clusters. We introduce two variants of Algorithm 1: a *full-WCDS* and a *bridged-WCDS*. We indistinctly use the terms dominator/reference and follower/dominee in the description below.

1) *full-WCDS*: In this variant, each follower maintains connections to all its neighboring references. In other words, a node that has  $k$  dominators listens to  $k$  different virtual channels. Figure 2(a) presents an example of a full-WCDS. As node  $C$  communicates with three dominators, it has to listen to all three virtual channels. It can be easily verified that all transmissions can use short preambles via the dominator/dominee links by construction.

2) *bridged-WCDS*: In this variant, to reduce the cost of listening to multiple virtual channels while maintaining connectivity, only a subset of common followers to two reference nodes acts as a bridge. Initially, each follower listens to exactly one virtual channel, which is the one maintained by the reference neighbor with the lowest id. Then, reference nodes elect bridge nodes. To do so, each reference node elects a bridge toward each reference node with a lower id and at most 2 hops apart. Thus, each node maintains the list of its 2-neighbors: each node includes the list of its neighbors and their states in its `hellos`. Then, a reference applies the MPR (Multi Point Relay) rule [12] to elect the bridges. It chooses at each step the follower that covers the maximum number of



(a) full-WCDS



(b) bridged-WCDS

Fig. 2. Examples of a full-WCDS and a bridged-WCDS.

reference nodes until it covers all of them. Figure 2(b) shows an example of a bridged-WCDS. Since the structure is still a WCDS, all the transmissions can benefit from short preambles.

## V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the original SCP, Crankshaft, and the two variants of WCDS (full-WCDS and bridged-WCDS) proposed in this paper. We consider energy consumption and network capacity. We simulate randomly generated networks with nodes that have 10 neighbors on the average (node degree). We use a unit disk communication model and plot 95% confidence intervals for all graphs.

### A. Energy Consumption

For the sake of conciseness, we neglect the energy drained in processing and sensing, and only consider the energy drained in communication. We use the default values of Wavenis nodes [5]: bandwidth=19.6kbps,  $\theta=20 \cdot 10^{-6}$ ,  $T_{CI}=1s$ ,  $T_{SI}=20min$ ,  $T_{SYNC}=0.012s$ ,  $T_{preamble}=0.096s$ ,  $P_{tx}=45mWatt$ ,  $P_{rx}=17mWatt$ , and  $\mathcal{P}_{samp}=32.51\mu Watt$ .

1) *Synchronization Maintenance and Sampling*: In the first experiment, we evaluate the cost of the protocols in an idle network, i.e. when there is no traffic to forward. We measure the mean power consumed in channel sampling and synchronization maintenance in a network of an increasing size (cf. Figure 3). As expected, SCP consumes the lowest power, because it uses a single virtual channel for the entire network. Bridged-WCDS achieves comparable power

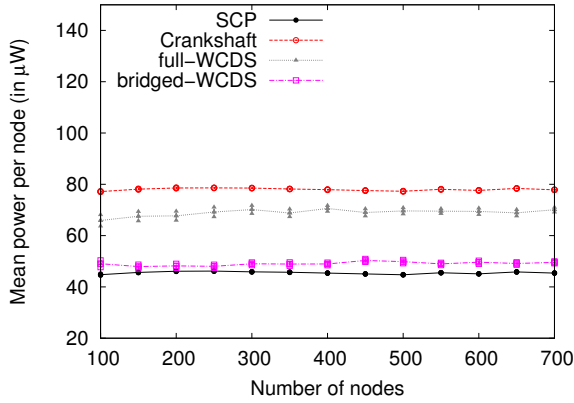


Fig. 3. Average power drained per node for synchronization and sampling in function of the network size.

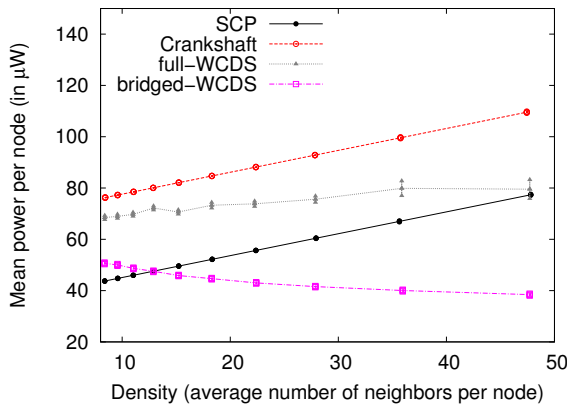


Fig. 4. Average power drained per node for synchronization and sampling in function of network density (node degree).

consumption to SCP while enabling multiple virtual channels in the network. It achieves this good trade-off by reducing the cost of synchronization maintenance. Full-WCDS consumes more energy than Bridged-WCDS, because follower nodes maintain more channels, but it further sustains connectivity. Crankshaft is the most consuming scheme, because each node maintains two channels. We can also see that all Synchronous Preamble Sampling schemes are scalable to large networks, because the mean power consumption remains constant when the network size increases.

We can see in Figure 4 that the mean power consumption of SCP and Crankshaft increases when the network becomes more dense. Actually, nodes have more neighbors and thus more references to listen to, which increases the synchronization cost. In contrast to that, the mean power consumption of full-WCDS remains constant. It even very slightly decreases for bridged-WCDS, because reference nodes are more uniformly distributed for higher density and slightly less bridges are required for interconnecting them.

2) *Traffic Forwarding*: In this experiment, we evaluate the overall energy consumption under convergeast traffic. We generate 10,000 packets (1 every 2 seconds) coming from different random sources to the sink. We measure the mean energy drained per node to transmit one packet. This includes

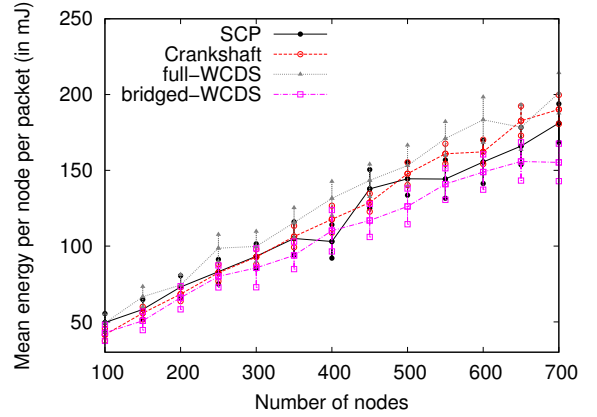


Fig. 5. Total energy per node drained in forwarding one packet in function of the network size.

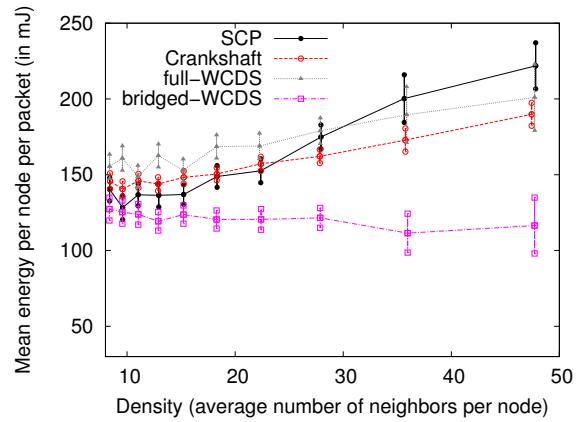


Fig. 6. Total energy per node drained in forwarding one packet in function of network density (node degree).

overhearing, channel sampling, synchronization maintenance as well as traffic forwarding.

Figure 5 shows that the energy consumption per node slightly increases with the network size. This is mainly due to border effects, because in a small network, nodes at the network border have less neighbors, thus the energy drained in overhearing is reduced.

Bridged-WCDS achieves the lowest energy consumption due to the minimization of synchronization maintenance cost and the reduction of overhearing through the use of multiple virtual channels. Note that energy savings can be even better if we use overhearing avoidance methods such as those proposed earlier [13]. As bridged-WCDS reduces the energy drained in synchronization and sampling, the energy savings compared to Crankshaft will be larger if the network has less packets to forward. Full-WCDS is not so efficient since it does not aim at minimizing the number of virtual channels per follower.

Although Crankshaft further reduces overhearing by using more virtual channels, it consumes more energy because of the costs of the additional broadcast channel and synchronization maintenance.

In Figure 6, we show the impact of network density on the mean energy consumption. The results confirm that bridged-WCDS and full-WCDS can efficiently save energy compared

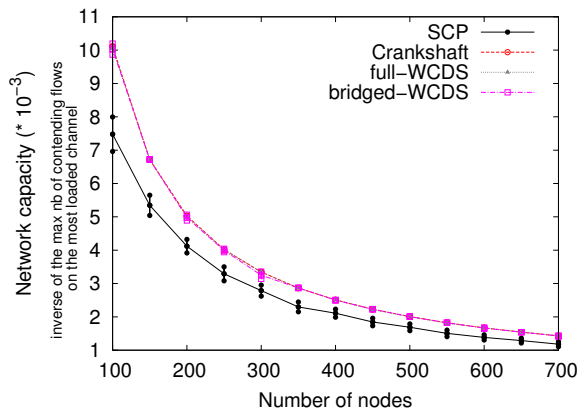


Fig. 7. Network capacity.

to the other schemes.

### B. Network Capacity

In a network with multiple virtual channels capacity increases, because nodes can forward more data with less contention. To estimate network capacity, we consider that all nodes send one unit of data toward the sink and define the capacity as the inverse of the largest number of interfering transmissions on the most loaded virtual channel. We find the bottleneck set of radio links in the conflict graph<sup>1</sup> of a given network. First, we associate a weight corresponding to the number of forwarded packets over each radio link. We assume that each node sends one packet to the sink. Then, for each virtual channel:

- 1) we construct the conflict graph for a given virtual channel, i.e. the conflict graph restricted to the links that use this channel.
- 2) we extract the maximum weighted clique in the conflict graph. An achievable scheduling in the max-clique is a necessary condition to obtain an achievable max-min bandwidth allocation<sup>2</sup> [14]. Since finding this clique is NP-hard, we approximate it with a greedy polynomial approach: we sort links by a decreasing weight and add a link to the clique if it interferes with all the links already in the clique.
- 3) we compute the total number of transmissions in this clique equal to the sum of weights of the links.

Finally, we estimate network capacity as the inverse of the number of transmissions over the most loaded clique (over all virtual channels).

Figure 7 presents the results of network capacity. We can show that Crankshaft, full-WCDS, and bridged-WCDS offer higher network capacity compared to SCP for any number of nodes, which confirms the benefit of using multiple virtual

<sup>1</sup>The conflict graph associates a vertex with each radio link. Two vertices are connected iff corresponding radio links interfere with each other. Although the conflict graph represents a binary interference model (transmissions either interfere or not), it provides a good approximation of realistic radio behavior.

<sup>2</sup>A max-min objective in multiframe networks means that we maximize the minimum bandwidth assigned to each flow. In other words, we *guarantee* a minimal bandwidth per flow.

channels. However, note that the difference decreases for large networks, because in this case contention appears around the sink even if nodes use multiple virtual channels.

## VI. CONCLUSION AND FUTURE WORK

We have proposed a new energy efficient network structure for maintaining synchronization in access methods based on Synchronous Preamble Sampling. Energy savings come from reducing the overhead of synchronization messages. Our scheme takes advantage of constructing independent clusters based on the Weakly Connected Dominating Set so that clusters can use different virtual channels and only need to maintain internal synchronization, while still offering global connectivity.

We have proposed two variants of the algorithm for distributed construction of clusters and compared their performance with main recently introduced access methods for wireless sensor networks. Our simulation results show that the proposed scheme has comparable energy consumption as Scheduled Channel Polling, but results in better network capacity. Moreover, it achieves better energy savings and network capacity than Crankshaft.

A direction for future research concerns finding the structure of the network that optimizes energy consumption for a dynamically varying traffic load instead of static structures.

## REFERENCES

- [1] A. El-Hoiydi, "Aloha with Preamble Sampling for Sporadic Traffic in Ad Hoc Wireless Sensor Networks," *IEEE ICC*, New York, NY, USA, April 2002.
- [2] J. Polastre, J. Hill and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," *In Proceedings of ACM SenSys*, 2004.
- [3] W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," *IEEE Infocom*, pp. 1567–76, New York, NY, July 2002.
- [4] W. Ye, F. Silva and J. Heidemann, "Ultra-Low Duty Cycle MAC with Scheduled Channel Polling," *ACM SenSys*, Boulder, CO, USA, November 2006.
- [5] "http://www.wavenis-osa.org."
- [6] J. E. Dunbar, J. W. Grossman, J. H. Hattigh, S. T. Hedetniemi, and A. A. McRae, "On weakly connected domination in graphs," *Discrete Mathematics*, vol. 167-168, pp. 261–269, April 1997.
- [7] A. El-Hoiydi, J-D. Decotignie, C. Enz, E. Le Roux, "Poster Abstract: WiseMac, an Ultra Low Power MAC Protocol for the WiseNET Wireless Sensor Networks," *ACM SenSys*, Los Angeles, USA, November 2003.
- [8] G. Halkes and K. Langendoen, "Crankshaft: An Energy-Efficient MAC-Protocol For Dense Wireless Sensor Networks," *EWSN*, 2007.
- [9] Y. Kim, H. Shin, H. Cha, "Y-MAC: An Energy-efficient Multi-channel MAC Protocol for Dense Wireless Sensor Networks," *In Proceedings of IEEE IPSN*, St. Louis, Missouri, April 2008.
- [10] D. Dubhashi, A. Mei, A. Panconesi, J. Radhakrishnan, and A. Srinivasan, "Fast distributed algorithms for (weakly) connected dominating sets and linear-size skeletons," in *SODA*. Baltimore, USA: ACM-SIAM, January 2003.
- [11] B. Han and W. Jia, "Clustering wireless ad hoc networks with weakly connected dominating set," *Journal of Parallel and Distributed Computing*, vol. 67, no. 6, pp. 727–737, June 2007.
- [12] A. Qayyum, L. Viennot, A. Laouiti, "Multipoint Relaying for Flooding Broadcast Messages in MobileWireless Networks," *IEEE HICSS*, Big Island, HI, USA, January 2002.
- [13] A. Bachir, D. Barthel, M. Heusse and A. Duda, "Micro-Frame Preamble MAC for Multihop Wireless Sensor Networks," *IEEE ICC*, Istanbul, Turkey, June 2006.
- [14] A. Karnik, A. Iyer, and C. Rosenberg, "Throughput-optimal configuration of fixed wireless networks," *IEEE/ACM Transaction on Networking*, vol. 16, no. 5, pp. 1161–1174, October 2008.