Energy-Efficient Broadcasts in Wireless Sensor Networks with Multiple Virtual Channels

Abdelmalik Bachir*, Fabrice Theoleyre†, Andrzej Duda†, and Kin K. Leung*
* Imperial College, London, UK
† Grenoble Informatics Laboratory, Grenoble, France

***Abstract*—Multichannel solutions are increasingly used to cope with the problem of low capacity in sensor networks resulting from high contention during wake-up periods of nodes. We can consider wake-up schedules as *virtual channels*, because nodes using different schedules cannot communicate with each other. In this paper, we show that multiple virtual channel solutions come at the cost of an increased energy consumption in broadcasts. We analyze the problem of broadcasting over multiple virtual channels under different classes of MAC methods and propose Clustered Virtual Channels (CVC), a new network structure that limits the number of frames needed for maintaining multiple virtual channels synchronized. Our simulation results show that CVC reduces the cost of maintaining synchronization and increases capacity while making all types of broadcasts possible.**

I. Introduction

Nodes in wireless sensor networks mainly save energy by maintaining their radio transceivers in sleep mode and periodically waking up for short periods to exchange data frames. To significantly save energy, the wake-up intervals need to be very short and their frequency low. Alternating sleep and wake-up periods requires coordination between nodes to avoid transmitting frames to a node during its sleep period. Much research effort has focused on developing MAC protocols achieving the goal of minimizing energy consumption [1] and the most efficient ones are based on *synchronous preamble sampling* [2]–[6] in which 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 [3]. To maintain the common schedule, each node needs to periodically broadcast a SYNC control frame to make up for clock drifts.

We can consider wake-up schedules as *virtual channels*, because nodes using different schedules cannot communicate with each other. In general, we can 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 [2] 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. Note that the classical definition of multiple channels usually refers to different channels at the physical layer (e.g. frequency bands, time slots, or chipping codes), which is a particular case of the virtual channel concept.

Operating with one virtual channel in the whole network as in Scheduled Channel Polling [3] presents many advantages especially for reducing energy consumption. However, it also suffers from some drawbacks essentially stemming from increased contention for accessing the single virtual channel, because nodes may only contend during very short wake-up periods. High contention results in high collision rate and low throughput. Besides, using the single common virtual channel makes overhearing possible so that nodes waste energy in receiving and processing frames for which they are not destinations. Many authors proposed solutions to deal with these issues based on the use of multiple channels, either virtual [4], [5] or physical ones [2], [6].

Using multiple channels in a sensor network reduces interference and collisions, improves network capacity, but also increases energy consumption. Indeed, a node that uses several different virtual channels needs to individually listen to each one of them, which consumes more energy compared to listening to one virtual channel. Moreover, maintaining virtual channels synchronized also requires periodic broadcasting of control SYNC frames to adjust the clocks of neighboring nodes. Thus, the overall network performance depends on the way multiple virtual channels are maintained and operated.

In this paper, we consider the problem of how to achieve energy-efficient broadcast communications over multiple virtual channels. Broadcast communications in wireless sensor networks are vital for network maintenance operations such as routing or neighbor discovery as well as for control commands or data dissemination. When the network relies on multiple virtual channels, achieving efficient broadcasts may be complex or difficult, for instance a node needs to know all the virtual channels used by its neighbors to reach them, which may be not always possible particularly in networks with varying topology.

In particular, our contribution is as follows:

- we analyze the energy costs of broadcast communications over multiple virtual channels under different classes of MAC methods,
- we propose Clustered Virtual Channels (CVC), a new network structure that limits the number of frames needed for maintaining multiple virtual channels synchronized,
- we show that the CVC structure also supports energy-efficient broadcast communications of different types,
- we evaluate the performance of broadcasts under different classes of MAC methods to show the advantage CVC.

II. Background

A. Energy Savings with Synchronous Preamble Sampling

Synchronous preamble sampling is the key technique for energy savings in wireless sensor networks [2]–[6]. Its most attractive feature is that it is not tailored to a specific configuration and thus can be used in a wide range of different topologies, traffic patterns, and applications. SPS saves energy by efficiently reducing idle listening, which is one of the major sources of energy consumption in sensor networks [7].

In synchronous preamble sampling, nodes save energy by sleeping most of the time and waking up at specific instants. The wake-up is periodic at each check interval to sample the channel. During a wake-up period, nodes consume much more energy than during sleep periods. Thus, the wake-up duration is kept to the minimum—just to detect whether there is an ongoing transmission or not to further reduce the energy drained in channel sampling. The wake-up duration is extended when a node detects the beginning of a transmission. It such a case, the node remains active until the full reception of the frame.

Using sleep periods along with active periods requires some coordination between nodes to enable communication (the receiver is listening to the channel while the transmitter is sending a frame), which means that each transmitter knows the virtual channel used by the corresponding receiver.

B. Capacity Improvement with Multiple Virtual Channels

Although using a single virtual channel reduces the complexity of synchronous preamble sampling, it has some drawbacks such as overhearing, decreased network capacity, and a potentially high collision rate. Contention can be particularly high under a single virtual channel, because all nodes contend for the channel during the wake-up period, which by design is very short.

To cope with these shortcomings, many authors propose to use multiple virtual channels. In Crankshaft [4], each node defines it own virtual channel on which it receives unicast data frames and an additional common virtual channel to receive broadcasts. The data virtual channels are assigned according to a simple scheme: each nodes determines the virtual channel according to its MAC address modulo the maximum number of virtual channels allowed in the network. In this way, a sender determines the virtual channel of the receiver based on its MAC address. Although such a technique reduces the overhead of unicast transmissions, it increases the overall energy consumption, because each node has to listen to the additional broadcast virtual channel. The overhead of listening to two different virtual channels (one for broadcast, the other one for unicast) may be significant especially for light traffic.

O-MAC [5] adopts a similar approach, but defines a more complex technique to obtain orthogonal virtual channels to further increase capacity and reduce collisions. Wavenis [2] and Y-MAC [6] follow the same idea with virtual channels based on different frequency bands or frequency hopping sequences to reduce interference.

III. Broadcast Communications

We start with making the notion of a broadcast more precise: we define below different types of broadcasts—depending on the goal of a broadcast, we distinguish three cases.

A. Neighbor Broadcast

This traditional notion of a broadcast consists in sending a given data frame to all known neighbors. In general, neighbors may use different virtual channels, so the sender needs to know all the virtual channels used by the neighbors to send a frame on each of them. Such broadcasting over multiple virtual channels may significantly increase energy consumption particularly when the number of virtual channels is large and the power used to transmit frames is high.

B. Neighbor Discovery

In this case, the broadcast consists in sending a given data frame to all known neighbors and to the nodes not yet known (e.g. the nodes that have just joined the network). The simplest way to reach the nodes for which we do not know the virtual channel is to have an additional virtual channel dedicated to neighbor broadcast transmissions common to all nodes as in Crankshaft.

C. Flooding

In this case, a node wants to send a given data frame to all the nodes in the network. This requires forwarding frames by nodes, which may consume much energy if all the nodes participate in forwarding. We observe that this type of broadcast communication is needed for sending SYNC frames to maintain network-wide synchronization if the network uses a single virtual channel and to maintain synchronized the dedicated broadcast channel in the case of multiple virtual channels (described below). Reducing the number of nodes that forward frames, which also means reducing the number of frames to transmit, can save energy.

IV. Maintaining Synchronization of Virtual Channels

When nodes coordinate their wake up schedules, they can use short preambles, which saves energy, but requires periodic transmission of SYNC frames to maintain synchronize clocks: a *reference node* transmits its clock in a SYNC frame and all receivers, i.e. the *followers*, adjust their clocks based on the information contained in the frame. Reference nodes need to broadcast SYNC frames so that they reach all followers.

Let us consider two main network structures from the synchronization point of view. To define energy costs, let us consider the following notation: E_{tx} (resp. E_{rx}) is the energy drained in transmitting (resp. receiving) a frame and E_{samp} is the energy drained in sampling (i.e. periodically listening to) one virtual channel. T_{CI} is the check interval, the period between two channel sampling, and T_{SI} is the synchronization interval, the period between two SYNC frame transmissions.

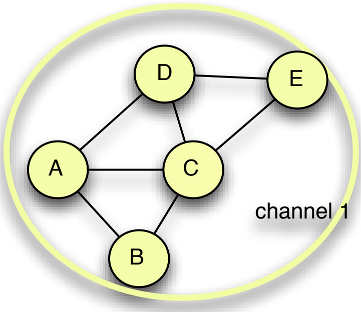

Fig. 1. SVC: Single Virtual Channel for all nodes

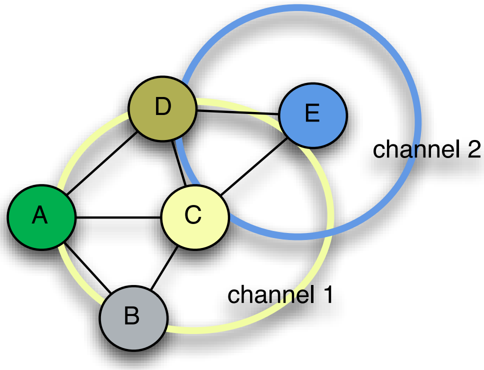

Fig. 2. MVC: Multiple Virtual Channels with a different virtual channel per node (only channels of nodes C and E are represented).

A. SVC - Single Virtual Channel

Figure 1 presents an example of a network in which all nodes use the same virtual channel. In this case, the overall energy cost is equal to the cost of maintaining synchronization plus the cost of channel sampling. The cost of maintaining synchronization corresponds to the cost of broadcasting one SYNC frame, which will be received by all neighbors, plus receiving the broadcasts of all neighbors when they forward their SYNC frames.

Let us take the example of node C in Figure 1. Node C has 4 neighbors, therefore it consumes a mean power equal to:

$$\frac{E_{\text{tx}} + 4E_{\text{rx}}}{T_{\text{SI}}} + \frac{E_{\text{samp}}}{T_{\text{CI}}} \tag{1}$$

B. MVC - Multiple Virtual Channels

In this case, which corresponds to the Crankshaft method, each node has its own virtual channel dedicated to receiving unicast frames (cf. Figure 2): node C listens to Virtual Channel 1 and node E to Virtual Channel 2 (only channels of nodes C and E are represented in the figure). If a node has frames to deliver to C, it must send them on Virtual Channel 1. Otherwise, node C will be deaf.

There is a dedicated channel for broadcast communications, which reduces the energy drained per transmitted broadcast frame. However, this gain is counter-balanced by the extra energy drained in sampling this channel. In this case, node C drains a mean power equal to:

$$\frac{E_{\text{tx}} + 4E_{\text{rx}}}{T_{\text{SI}}} + 2\frac{E_{\text{samp}}}{T_{\text{CI}}} \tag{2}$$

We can observe that the dedicated broadcast channel comes at a large cost, because it doubles the power drained in sampling, which is the dominant term especially in low traffic sensor networks.

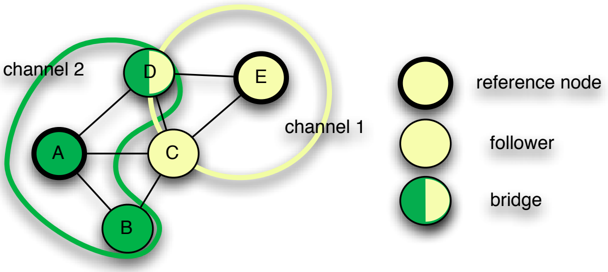

Fig. 3. Clustered Virtual Channels: a virtual channel per cluster of nodes.

V. Clustered Virtual Channels

To enable energy-efficient broadcasts in multiple virtual channel networks, we propose Clustered Virtual Channels (CVC), an improvement to bridged-WCDS, the structure proposed previously [8] and tailored to reducing the overall energy consumption in sensor networks by limiting the number of transmitted SYNC frames. Bridged-WCDS is based on the notion of WCDS (Weakly Connected Dominating Set) [9]. In a sensor network represented as an undirected graph $G = (V, E)$, where V is the set of nodes and $E \subseteq V^2$ is the set of edges (edge (u, v) belongs to E iff u can communicate with v), a WCDS is a set of nodes satisfying the following conditions:

$$\forall d \notin \text{WCDS},\ \exists d' \in \text{WCDS} | d' \in N(d) \tag{3}$$

$$G(V, E') \text{ connected} | E \supseteq E' = \{(u, v)\}, u \in \text{WCDS}, v \in V \tag{4}$$

where $N(u)$ is the set of neighbors of node u.

The nodes belonging to the WCDS set are called *dominators* and the other ones *dominatees*. From the synchronization point of view, dominators are time references i.e. they periodically broadcast SYNC frames, while dominatees are followers: they receive SYNC frames and adjust their clocks accordingly. A dominator and the set of its neighboring followers form a *cluster* with the dominator being its *clusterhead*. There is a virtual data channel common to all nodes of a given cluster derived from the MAC address of the clusterhead: all followers know it from transmitted SYNC frames. Figure 3 shows an example of a WCDS structure. There are two dominator nodes A and E with their corresponding clusters $\{A, B, D\}$ and $\{E, D, C\}$.

The WCDS structure reduces energy consumption, because only dominators need to periodically broadcast SYNC frames as opposed to forwarding by all nodes required in traditional methods. Besides, it improves the capacity by using several virtual channels. Indeed, each cluster uses its own virtual channel for all transmissions internal to the cluster. A reference node always listens to one single virtual channel (that of its cluster). However, to form a WCDS, some nodes must be members of several clusters to form a connected set (e.g. node D is a member of the clusters of A and E in Figure 3). Consequently, such a node has to listen to all virtual channels of the clusters it belongs to (e.g. channels 1 and 2 in Figure 3).

A. *Enabling Energy-Efficient Broadcasts*

In our approach, nodes easily achieve Neighbor Broadcasts, because a node can send its broadcast frame on all virtual channels of its neighbors.

To enable Neighbor Discovery broadcasts, nodes use a network-wide virtual channel called *discovery broadcast channel* and maintain synchronization through periodic exchanges of SYNC frames. In CVC, we let bridge nodes participate in flooding SYNC frames instead of being passive followers. More specifically, each follower floods a SYNC frame it receives to all reference nodes for which it acts as a bridge. Reference nodes adjust their clocks to the oldest received clock so that step by step the network synchronizes to the common clock. Because of the WCDS structure, if a bridge node forwards the clock of its reference nodes it follows, we will achieve a network-wide synchronization.

By default and when there is no traffic, the broadcast virtual channel is only active every synchronization interval T_{SI} for exchanging SYNC frames. It can be active more often, but this implies additional energy spent in sampling, which is not desirable in the absence of discovery broadcast traffic. Ideally, nodes need to dynamically change their sampling interval according to the load of broadcast transmissions to continuously achieve optimal energy savings.

When all nodes have a common time reference, transmitting Neighbor Discovery broadcasts drains less energy, because only reference nodes and bridges need to transmit broadcast frames. This significantly reduces energy consumption compared to previously proposed solutions [3], [4] in which all nodes need to flood SYNC frames.

Finally, flooding can benefit from the WCDS structure: a node sends a frame to its clusterhead that broadcasts it on the cluster channel and bridges forward it further to neighbor clusters. We will explain this process after a detailed presentation of CVC.

B. *Energy Cost of CVC*

Nodes can save energy with the dedicated broadcast channel, because the cost of transmitting a broadcast frame reduces to the cost of transmitting a single frame and the cost of sampling is reduced to sampling one data virtual channel and one broadcast virtual channel. In the example of Figure 3, node C drains a mean power equal to:

$$\frac{E_{\mathrm{tx}} + 4E_{\mathrm{rx}}}{T_{\mathrm{SI}}} + \frac{E_{\mathrm{samp}}}{T_{\mathrm{CI}}} + \frac{E_{\mathrm{samp}}}{T_{\mathrm{SI}}} \qquad (5)$$

The drained power is significantly smaller than that of Eq. 2, because T_{SI} is typically much longer than T_{CI} (cf. Table I for typical values of the parameters).

TABLE I
VALUES OF PARAMETERS USED IN SIMULATIONS

SYNC frame size	30 bytes
Short Preamble maximum duration	0.096 seconds
Bit rate	19.6 kbps
Check Interval	1 seconds
Synchronization Interval	20 minutes
Transmission power	45 mWatt
Reception power	17 mWatt
Energy drained in sampling	$10,84$ μJoule

C. *Efficient Broadcast and Unicast Transmissions*

a) Neighborhood discovery: A node periodically exchanges `hello` frames with the other nodes in its radio range. To achieve this, it uses the discovery broadcast channel. It can send one single hello frame to reach all its neighbors regardless of their clusters (i.e. virtual channel used in a cluster). Besides, a node includes in its `hellos` the list of cluster ids it belongs to, which also represents the set of virtual channels it listens to.

b) Unicast transmissions: As explained previously, a node uses a *data channel* for all its unicast transmissions. A node has just to extract from its neighborhood table the virtual channel used by the destination node and to use it for its unicast transmissions toward this node.

c) Neighbor broadcast: A node may need to reach all nodes in its radio range. If the transmission is not delay sensitive, it can use the discovery broadcast channel. However, the delay may be too long for some control packets (e.g. a *route request*). Thus, such a node has to replicate the packet and to send one copy per virtual channel of its known neighbors.

d) Flooding: Nodes have sometimes some information to flood in the whole network e.g. routing updates, code or content dissemination etc. A node can use the broadcast virtual channel for its delay-tolerant broadcasts, because it may adjust sampling periodicity and adapt to varying broadcast load. On the contrary, for broadcasts that require small delays (i.e. in $O(T_{CI})$), nodes can use the data virtual channel. The situation depends in this case on the type of the node transmitting a broadcast frame:

- a reference node (e.g. node A in Figure 3): broadcasting is straightforward, the reference node just transmits a frame on the virtual data channel of its cluster. Indeed, all its cluster members will receive the frame, and other neighbors will receive it indirectly because of the WCDS global connectivity;
- a bridge (e.g. node D): it needs to transmit as many broadcast frames as clusters it is connected to. For instance, node D forwards a frame to both A and G on two different virtual channels;
- a follower (e.g. node B): it just has to transmit a broadcast frame to its neighbors. For instance, node B just sends the frame to A and C over their virtual channels.

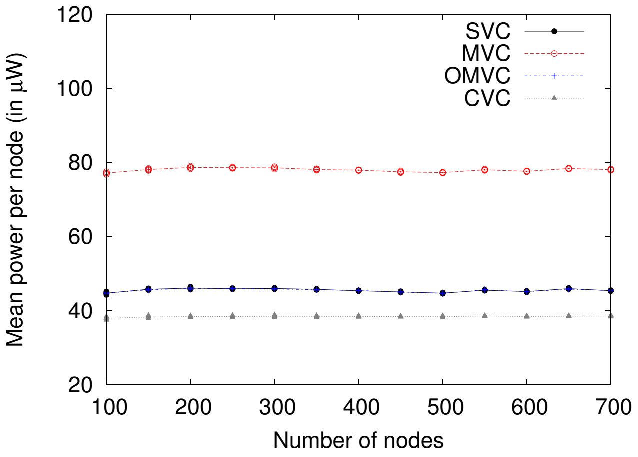

Fig. 4. Average power drained per node for synchronization and sampling in function of the network size.

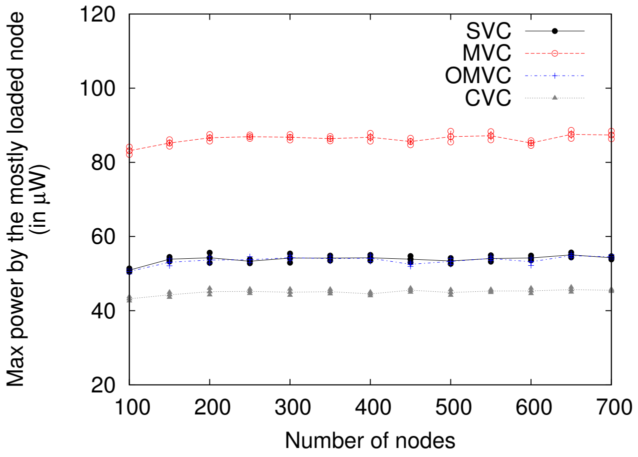

Fig. 5. Average power drained by the most loaded node for synchronization and sampling in function of the network size.

VI. Performance Evaluation

A. Simulation Configuration

We compare the performance of our CVC approach with two other ways of transmitting broadcasts: over a single virtual channel (SVC) and over multiple virtual channels (MVC). To be fair, we also consider an optimized version of multiple virtual channels (OVMC) in which the broadcast virtual channel is *partial-time* as opposed to the *full-time* virtual channel of Crankshaft [4]. We focus on evaluating energy consumption and network capacity, the metrics we consider as the most important in sensor networks. We also discuss the issue of robustness related to the overall connectivity in the network.

In our experiments, we simulate randomly generated networks in which each node has 10 neighbors on the average. We have used typical values presented in Table I corresponding to the specification of Coronis Systems products [10]. For all the graphs plotted below, we present 95% confidence intervals.

B. Energy Consumption

For the sake of conciseness, we neglect the energy drained in processing and sensing, and we only consider the energy drained in radio communications. We measure the energy drained in channel sampling as well as in frame transmissions and receptions.

We first evaluate the energy drained for maintaining synchronization and sampling. Figure 4 presents the mean power consumed by an average node. It shows that CVC consumes the least power for structure maintenance, because it uses an optimized network-wide synchronization maintenance that relies on broadcasting rather than on flooding. We can see that although CVC uses an additional discovery broadcast channel, it consumes less than SVC that only uses one channel, because the cost of using the additional discovery channel is minimal. The difference between using full-time and partial-time broadcast channels can also be seen from the difference between MVC and OMVC. The results obtained for an average node also hold for the most loaded node as shown in Figure 5.

Figures 4 and 5 also show that all the protocols are scalable, because the mean energy consumption remains constant for an increasing network size.

C. Network Capacity

To estimate network capacity, we proceed as follows. When nodes use multiple channels, network capacity increases, because they can handle more data with less contention. We neglect interference between different virtual channels. We consider that each node sends one unit of data toward the sink and we estimate network capacity as the inverse of the largest number of interfering transmissions on the most loaded channel. Thus, we construct the conflict graph for each existing virtual channel and extract the bottleneck from it, i.e. the clique of interfering radio links that must forward the largest amount of traffic. Finally, we compute the capacity as the inverse of the number of transmissions in this clique.

Figure 6 presents network capacity. We can note that SVC provides the lowest capacity, because it uses a single channel, which increases contention between nodes and does not allow concurrent transmissions. Moreover, CVC provides more capacity than SVC, but less than MVC: MVC allocates one channel for each node whereas CVC assigns one channel for each cluster (one reference node and its neighboring followers).

Note that although Figures 4, 5, and 6 suggest that OMVC obtains good performance, it is not as efficient as CVC. Indeed, the channel dedicated to broadcasts does not allow to forward delay-sensitive broadcasts except if we duplicate broadcast frames on each virtual channel. Recall that in OMVC, a node either waits for the next available broadcast slot (one every synchronization interval) or performs broadcasts as several unicast transmissions (one unicast per neighbor), which either generates a long delay or consumes a large amount of energy.

D. Robustness and Network Connectivity

Network connectivity is an important aspect of sensor networks. In a SVC network, the issue is somehow easier

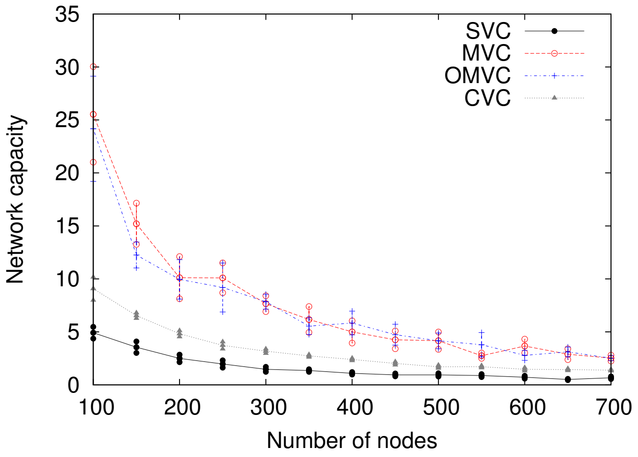

Fig. 6. Network capacity in function of the network size.

to address, because a node can directly communicate with any of its neighbors. However, in multichannel approaches, some neighbor nodes may be not reachable, because they use different virtual channels. In all the structures considered in this paper, the network stays connected, because all nodes are synchronized and the channel used by a node can be inferred either from its MAC address such as in MVC or from the MAC address of the dominator to which it belongs as in CVC.

VII. Conclusion and Future Work

Low capacity of single channel sensor networks calls for MAC protocols based on multiple channels. However, multichannel approaches may also increase energy consumption, because a node needs to sample several virtual channels. Moreover, maintaining virtual channels synchronized also requires periodic broadcasting of control SYNC frames to adjust the clocks of neighboring nodes.

In this paper, we have considered the problem of how to achieve energy-efficient broadcast communications over multiple virtual channels. We have proposed Clustered Virtual Channels (CVC), a new network structure that builds on the bridged-WCDS to establish clusters in the network, each cluster using a different virtual channel to allow efficient intra cluster broadcasting while increasing capacity by allowing concurrent inter-cluster transmissions. The results of simulations show the efficiency of the bridged-WCDS structure. In the future, we aim at finding optimal scheduling for network-wide broadcasts with the minimum cost, dynamically changing clusters, and the check interval of the discovery broadcast virtual channel to meet varying traffic load.

References

[1] K. Langendoen. Energy-Efficient Medium Access Control. *Book chapter in Medium Access Control in Wireless Networks, H. Wu and Y. Pan (editors), Nova Science Publishers*, 2008.

[2] http://www.wavenis-osa.org.

[3] W. Ye, F. Silva and J. Heidemann. Ultra-Low Duty Cycle MAC with Scheduled Channel Polling. *ACM SenSys*, Boulder, CO, USA, November 2006.

[4] G. Halkes and K. Langendoen. Crankshaft: An Energy-Efficient MAC-Protocol For Dense Wireless Sensor Networks. *EWSN*, 2007.

[5] H. Cao, K. Parker, and A. Arora. O-MAC: A Receiver Centric Power Management Protocol. *In Proceedings of IEEE ICNP*, Santa Barbara, CA, November 2006.

[6] Y. Kim, H. Shin, and H. Cha. DY-MAC: An Energy-efficient Multi-channel MAC Protocol for Dense Wireless Sensor Networks. *In Proceedings of IPSN*, pages 53–63, St. Louis, MI, April 2008.

[7] W. Ye, J. Heidemann, and D. Estrin. An Energy-Efficient MAC Protocol for Wireless Sensor Networks. *IEEE Infocom*, pages 1567–76, New York, NY, July 2002.

[8] F. Theoleyre, A. Bachir, N. Chakchouk, A. Duda, and K. K. Leung. High-Capacity and Energy-Efficient Virtual-Channel Assignment for Sensor Networks. *Submitted to ICC*, 2010, http://membres-liglab.imag.fr/theoleyre/tmp/icc10-submitted-virtual_channel_assignment.pdf.

[9] Jean E. Dunbar, Jerrold W. Grossman, Johannes H. Hattingh, Stephen T. Hedetniemi, and Alice A. McRae. On weakly connected domination in graphs. *Discrete Mathematics*, 167-168:261–269, April 1997.

[10] http://www.coronis-systems.com.