A Self-Organization Mechanism for a Cold Chain Monitoring System

Charbel Nicolas, Michel Marot and Monique Becker
CNRS-SAMOVAR-UMR 5157, Département-RST
Institut TELECOM:TELECOM SudParis
9 rue Charles Fourier, Evry, France
Email:{ Charbel.Nicolas, Michel.Marot, Monique.Becker}@telecom-sudparis.eu

Abstract — **In this paper, we propose an autonomous self-organization mechanism for the monitoring of the cold chain. All along the transportation through this logistic chain, the sensors are moved with the goods in very different networks. We argue that depending on the situation some protocols are more suited than others and it is necessary the sensors to adapt and to switch dynamically depending on where they are. The mechanism we designed allows a sensor to autonomously detect the context where it is and to choose the best protocols for this situation. It assures the adaptability to different contexts where a sensor may move (warehouses, trucks,...). It implements a hybrid synchronization mechanism to assure duty cycle synchronization between nodes with high power efficiency and context adaptation. This hybrid mechanism uses wake up signals to activate a sleeping sensor to assure fast synchronization and efficient power consumption optimization. This mechanism also changes dynamically the routing protocol to the different networks the sensors enter in (dense or sparse topologies, with or without total visibility among all nodes,…).**

Keywords-WSN; CSMA/CA; TDMA; passiveRFID; Routing; Duty cylcling; PLACIDE; MAXMIN;Cold chain monitoring;

I. INTRODUCTION

A cold chain is a temperature-controlled supply chain. It is an uninterrupted series of storage and distribution activities which maintains a given temperature range. Cold chain is an important part in delivering perishable goods especially medicine, food, chemical products etc. It involves multiple actors, from warehouses to retailers, but also trucks and trains or maritime transportations and it involves generally multiple responsibilities. We are interested in an end to end cold chain monitoring system. In our case there are two requirements. The triggered alarms are not checked immediately: the system is used only to detect and record the responsibility of the guilty actor in the cold chain. Therefore the wireless sensor network (WSN) monitoring the cold chain does not need to use real time mode to broadcast an alarm, rather than that it is stored in the sensor's memory for future transmission. Moreover, in our work, no infrastructure and base station must be used to collect the data because the logistic actors are too preoccupied with the increase of the price of the petroleum and logistic transportation specific costs to be ready to accept an additional infrastructure expense. The data are then broadcast to all sensors assuring a consistent alarm database between all the sensors of the WSN. At any time, all the data can be queried with a simple PDA from any sensor.

There are solutions to monitor the cold chain (cf. [1] for example), but it generally consists in having data loggers in each part of the cold chain, trucks, trains, warehouses, etc., without any end to end networking facility. Several data loggers belonging to different juridical actors in the chain cannot give a transparent, independent and global view of the chain to monitor the responsibilities. In [2] a solution based on sensor networks is proposed but it requires an infrastructure, which has a prohibitive cost in practice. In [3] another solution is proposed but the authors do not address the scalability issue, esp. in large warehouses. Also, in these solutions, the data are not broadcast over the nodes to be retrieved later from any node.

A solution exists which is not end to end, which satisfies the constraints of the event data bases to be the same on all the sensors and to use no infrastructure: the PLACIDE [4] protocol. It is not end to end but it is well suited to the specific portion of the chain constituted by the truck transportation. This portion has very specific properties: all the sensors are in total visibility, though there may be wave obstacles like ices or frozen goods, because the pallets are in the same plane on the floor of the truck (the sensor are put inside the pallets) and they are less than 33 because of the surface limitation of the truck. In this case it can be proved that PLACIDE is the optimal protocol in terms of energy consumption. It is a mechanism which synchronizes the sleeping, wake up and transmission times of the nodes and it allows the data to be broadcast between the nodes through a logical ring between the nodes. It is optimal because it lets the data be broadcast through the path they follow (the ring) which covers all the nodes but it necessitates the node to send and hear only once per duty cycle. In the warehouses, where PLACIDE cannot be used because of lack of total visibility, other solutions must be implemented. In these case there are so many nodes that a multi-hop cluster based routing protocol is indispensable. This allows having several "PLACIDE-like" sub-networks in parallel. MAXMIN (cf. [6] and [7]) is a good heuristic to build multi-hop clusters, the data being broadcast then between the clusters since we want all the sensors to have the same data base.

Since these solutions are not end to end, but optimized for each part of the chain, it is necessary to have a mechanism able to autonomously detect the part in which a sensor is and to apply automatically the proper protocol depending on the detected context. In this paper we propose such a mechanism that we call Context Aware Mechanism (CAM) in the sequel. Basically, for each sensor it allows to autonomously detect if it is inside a truck or inside a warehouse and it dynamically switches between PLACIDE and a cluster based routing mechanism using MAXMIN (by extension this routing algorithm will be shorten as MAXMIN). Let us recall that our logistic client wants to have no infrastructure at all, which makes the detection of the context quite difficult. Thus it is not possible to install base stations in the truck or warehouses which would announce where the sensors are.

Moreover, the detection of the context is pertained to the synchronization mechanisms of the nodes. Actually, a node detects a context change when its neighbors have changed, that is when a node ceases to transmit at the time slot it should or when a new node just arrives. When PLACIDE is used, the method proposed by PLACIDE is reused, when it cannot (in the warehouses), this method is extended and adapted to the tree structure of the clusters. For our goal it is important to have a synchronization method which is compatible in both cases. That is why the mechanism we propose is not only a self-organization mechanism allowing to switch from PLACIDE to a cluster based routing algorithm but also an extension of the PLACIDE synchronization protocol to the clustered network part which synchronizes the nodes and which plans the sleeping, wake up and transmission times. For the slot allocation, the method resembles Z-MAC [5] except that we forbid several nodes to locally transmit in the same slot and it resembles DRAND [10] except that nodes are not required to be always awake or periodically synchronize.

The next parts describe PLACIDE and then the mechanism by starting with the synchronization part followed with the routing part. A performance evaluation is then presented and the last part is dedicated to our conclusion.

II. RELATED WORK: PLACIDE

PLACIDE allows the data to be broadcast between the sensors and it optimizes the wake up and sleeping periods. It sets up a loop between the sensors. The wake up periods are extremely small (a few milliseconds) and the information sent by the sensor number n to the sensor number n+1 in the loop is sent by the sensor n+1 to the sensor n+2 together with the information sensed by the sensor n+1 so that the information is actually broadcast through the ring. Moreover, it has been designed to be used only in the trucks where all the sensors are in total visibility and where they are at most 33. As it synchronizes the nodes in a loop in which the information flows following to the direction of the loop, it works as if it were a cross-layer MAC and routing protocol (it is in fact an applicative protocol on top of the IEEE 802.15.4).

The construction of the logical ring network is done during an initialization phase, which belongs to the synchronization part of the mechanism, where each node advertizes its presence by using an ALOHA mode. The order of the nodes in the ring is the order of the emission times of the advertisements. If there is a hidden terminal: multiple rings may be created at the same time causing collisions and instability of the network.

In our case, we must thus modify the synchronization phase to check that all nodes are in total visibility. Moreover, a synchronization process is needed when MAXMIN, which only defines the path the information should be routed through, is used. That is why we separated the synchronization part of PLACIDE from its routing one and designed a new synchronization process common to PLACIDE and MAXMIN. In particular, during the synchronization process, and contrary to PLACIDE, the CAM lets the nodes stay awake to have time to detect all their neighbors, and in case of hidden terminals or a number of nodes greater than 33 CAM chooses another more suited routing protocol (MAXMIN).

III. DESCRIPTION

CSMA/CA without RTS/CTS is used to perform the signaling, and no acknowledgment after packet reception to minimize the energy consumption. The data packets are normally sent in a TDMA mode that is why each node has, and exchange with its neighbors, a scheduling table containing for itself, for each neighbor and for each neighbor of neighbor the identifier of the node, the transmission time, the chosen routing algorithm and the cluster head. The CAM detects the context from detected modifications of this information. Two hop neighbor information is used first to ensure that there is a total visibility between all the sensors and then that PLACIDE can be applied (when there are less than 33 neighbors), and second to communicate with a new sensor without causing collision with the neighbor's neighbors. All this exchanged information is thus also used by a new sensor to calculate its own listening and transmission times.

After a context detection and synchronization, to determine if PLACIDE must be used instead of a MAXMIN cluster configuration, the CAM counts the number of neighbors to check if it is less than 33 and, if it is the case, it compares for each neighbor node X, from its scheduling table, the set of the neighbors of this neighbor X to the set of its own neighbors, to check if it has the same set of neighbors as X. If it is true, there is a total visibility between all the neighboring sensors and then PLACIDE is applied, otherwise MAXMIN is used. For example, on the network of the left of Figure 1 node 1 as the same neighbors as 2, 3 and 4, so they are all in visibility, but on the right, 4 and 1 have different sets of neighbors.

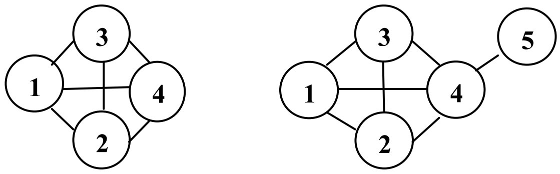

Figure 1: Different neighboring situations

A. *Synchronization and topology change detection*

1) *What is "synchronization"*

The mobility and the duty cycling generate a heavy signaling for the synchronization, that is why we choose to use a out-of-band wake-up channel, on a different specific frequency, using a passive circuit tag to trigger a wake up signal to start a synchronization when a topology change is detected. It is any electronic circuit which can be used to trigger an external interrupt to the microcontroller allowing it to wake-up its radio module to receive data (e.g. passive RFID, cf. [8]). At every wakeup signal, the radio of all neighboring sensors is turned ON and waits for a request. After this request is transmitted the neighboring nodes reply and then an acknowledgment from the requesting node finishes the signaling period (cf. Fig.2).

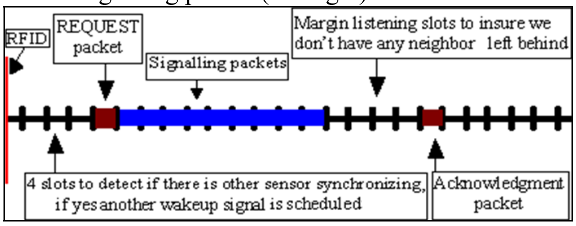

Figure 2 : After a wakeup signal events time line

What we call synchronization is an agreement on timeslots where a sensor transmits and receives data without causing collisions and disturbing the network. During the synchronization a number of actions should be taken:

1. After the wake up signal, the neighboring sensors enter in signaling mode using CSMA/CA. This mode does not end until the new sensor sends its transmission time. The new sensor requests from the neighbors their scheduling tables.
2. At the reception of these tables the sensor chooses a timeslot not already chosen by the others and it broadcasts it to the neighbors as its permanent transmission time.
3. During this message exchange each sensor affected by the wakeup signal listens to the messages sent by its neighbors and updates its own tables.
4. After that, the signaling ends and, in the next cycle, the scheduling update is forwarded to the neighbors of neighbors.

2) Means to detect a new sensor in a network

There are two cases allowing to detect a change of context causing a synchronization request to be sent: the detection by a node that the number of its neighbors has changed and a data packet collision detection. Then, the CAM triggers a wakeup signal to calculate a new transmission time.

A Node listens to its neighbors to collect their messages, and if it has not received any message from any neighbor, it tries to listen to all its neighbors that exist in its scheduling table (i.e. after applying the routing algorithm some neighbors are flagged so that the sensor does not listen to all neighbors, because the same information is received by another neighbor) and then, if less than 50% of neighbors reply, it triggers the wakeup signal and starts a new synchronization phase. If there is no answer for a request to synchronize there is another wakeup signal after a certain time.

The first time there is a collision, a flag is razed but no action is taken to ensure that it is not caused by a new neighbor that is not synchronized yet. After one cycle if there is a second collision a wakeup signal is triggered followed by a resynchronize request indicating that the nodes that transmitted at the flagged time slot should send their scheduling table. After, the neighbor that has the lowest number of neighbors (to ensure a minimal energy cost) is requested to resynchronize. In the case where there is more than two replies to the scheduling table request this means that there is more than two neighbors that are involved in the collision, then only the neighbor that has the lowest number of neighbors is not affected by the resynchronization request.

If a sensor does not receive from a neighbor a packet after two periods it eliminates the neighboring sensor from its table and it considers it as extracted from the network.

B. Choosing the routing algorithm

Once synchronized, the sensors are ready to choose the best routing algorithm depending on the context. Since PLACIDE requires first a total visibility between the sensors and second a limited number of sensors, less or equal to 33 sensors, in order to apply PLACIDE the sensors check in their tables if each sensor has the same neighbors, that is if a total visibility is assured, and if their number is less than thirty three. When a sensor finds these conditions verified, it decides to apply the PLACIDE protocol, otherwise it uses MAXMIN.

1) Node insertion and extraction in a network using PLACIDE

Any new sensor insertion into the network managed by PLACIDE is followed by the same tests and comparisons. If the conditions to apply PLACIDE are satisfied there is no problem because all the neighbors including the new one are synchronized.

If it is not the case, the procedure to apply MAXMIN starts. First the sensor synchronizes itself with its neighbors that are already using PLACIDE, causing a change in their tables, at the same time the two hop neighbors are updated. Each sensor in its own checks the conditions to continue to apply PLACIDE. If one of the PLACIDE constraints is broken, all the sensors in the network using PLACIDE directly converge to one routing algorithm choice: MAXMIN. After the insertion of the new sensor, the part of the network that is not affected by the wake up signal triggered by the new sensor is still using PLACIDE while the other part has just chosen MAXMIN. This does not cause the network to collapse because the connectivity and the synchronization of the sensors continue to exist but the sensors that detected the change directly stop applying PLACIDE and start to listen to all the neighbors waiting to confirm the start of the MAXMIN signaling. When all the nodes detect that their neighbors have changed their routing to MAXMIN, the MAXMIN signaling starts and the information exchange to choose the appropriate cluster head begins. The confirmation to apply MAXMIN comes from the first sensor which chooses MAXMIN.

In the case of PLACIDE, when a node decides to switch from another protocol to PLACIDE, the choice is immediately applied because, all the nodes being in total visibility, they all receive from the new node the new scheduling plan, but in the case of MAXMIN it is more complicated because there may be a lot of sensors that should exchange signaling messages, and calculate the cluster head, and therefore the same period length as the duty cycle (ex: 20 minutes) to propagate the choice cannot be used: it would need a long time to converge. Thus, when MAXMIN is chosen by the nodes, the sensors do not come back to the duty cycle period but they obligate the neighbors to stay awake and communicate using CSMA/CA during a certain time to propagate the choice.

After a confirmation reception to apply MAXMIN is received, they remain awake and calculate the MAXMIN table. For that, they exchange the MAXMIN weights (the criteria used for the choice of the cluster head) in CSMA/CA. After four idle timeslots, if no message is received the sensor requests the weight if needed. Then, the sensor compares the weights and if a sensor discovers that it is a cluster head it triggers a flood till a certain number of hops to declare itself and construct the cluster. Then, each sensor has received and transmitted the cluster head choice, the sensor returns back to its synchronized state scheduled in its scheduling table.

In the case where a sensor is extracted from a network managed by PLACIDE, it does not affect the routing choice, only the wakeup of the successor after the extracted node in the loop: rather than listening to the sensor that is detached the successor listens to the one that is, in the schedule, before the extracted node, and then it informs all the neighbors to update their tables and remove the extracted sensor. The duration to propagate the extraction is one cycle.

2) Node insertion and extraction and periodical clusterhead reelection in a network using MAXMIN

After a wakeup signal from a new sensor the signaling mode starts with the neighbors. After synchronization, the new sensor becomes connected. When the network is using MAXMIN whatever is inserted into the network it stays controlled by MAXMIN because there is already a broken constraint of PLACIDE. If a network is managed by MAXMIN the only case to change to PLACIDE is when sensors are extracted and the PLACIDE constraints satisfied. When the new sensor is synchronized, any decision to change the cluster head in the cluster and the reconstruction of the cluster is only made by the cluster head itself.

The extraction of a sensor is more complicated because there may be big changes like the extraction of a cluster head. When the cluster head is extracted, its neighbors detect its disappearance after two cycles from the time at which the cluster head did not transmit anymore. Then, its neighbors act as if they had received a message from the cluster head to reconstruct the cluster. Usually each message sent by the cluster head is tagged by it and forwarded through a cluster. Thus, if the cluster head is gone, all the cluster members know its disappearance and the signaling should start after.

Any extraction of a regular sensor causes a test to apply PLACIDE by its neighbors. If PLACIDE should be applied, a message is sent to the cluster head giving the order to change to PLACIDE and it is directly applied. In case of an extraction of a neighbor that put on hold by some of its neighbors after executing MAXMIN routing, its extraction is detected and sent to the cluster head that propagates the extraction so that its neighbors remove it from the scheduling table.

At last, periodically within a MAXMIN network, MAXMIN is run again to refresh the cluster head choices. When elected, the cluster heads broadcast a message to ask MAXMIN to be run again after a certain period.

IV. ENERGY CONSUMPTION AND SIMULATION RESULTS

To evaluate our mechanism, we first consider the energy consumed by the different parts of our mechanism (synchronization signal overhead, MaxMin signaling overhead and data transmission) to check if it is compatible with the order of magnitude of the energy stored in typical batteries. But, since the synchronization delay is also an important performance criterion of our system because of the mobility inherent to the cold chain, we also look at the synchronization delay in the trucks which is the highest synchronization delay in the network.

We built in C our own discrete event simulator to obtain a first evaluation of the performance of our proposed mechanism. To model the pallet transportation, 33 pallets are randomly and uniformly pushed from the warehouse and put together in a truck. At the same time, two timers exponentially distributed are triggered: one modeling the time when the truck will come back to the warehouse and another one modeling the next truck departure. When the truck comes back, the 33 pallets are randomly and uniformly put in the warehouse. After synchronization each sensor transmits 1packet/20min and receives possibly one or several packets from its neighbors depending on PLACIDE or MAXMIN.

The parameters used in the simulation are:

1. The link capacity is 250Kbps.
2. The length of each packet is 132Bytes.
3. Each period is 20 minutes (282352 slots).
4. The mean of the exponential time between two arrivals is 4 hours and the same for the mean of the exponential time separating two truck departures.
5. The coverage radius of the wakeup signal is the same as data transmission range: 6m.
6. The area where the nodes are gathered in the warehouse is 6000m2 and for the vehicle 12 m^2.

A. *Evaluation of the consumed energy*

We evaluated the number of packets exchanged by the total number of nodes, and then we deduced an estimate of the energy consumed by a node. The energy cost (*E*) to transmit or receive a packet is equal to 0.027795 mJ.

The number of triggered wakeup signals depends on the mobility of the neighboring nodes. The consumed energy related to a wakeup signal is equal to: $E_{WS}= (N*E_{Trx}) + E_{RFID}$. E_{RFID} is the energy consumed to trigger a passive RFID signal equal to 1.2mJ (cf. [9] for an example of RFID signal energy calculation) for both the receiver and the transmitter. E_{Trx} is the energy consumed during the signaling message exchange between neighbors, which is equal to the request, the acknowledgment packet and 8 margin slots plus the neighbor's packets. N is the number of affected neighbors.

E_{WS}=N*1.2*10e-03+ (N+10)*E.

On Figure 3 is plotted the energy consumed during two years by the network for the synchronization signaling, for the data packet transmission and for the MAXMIN signaling. It is noticeable that the synchronization cost is quite low. Naturally, it depends on the frequency of the changes and it may be larger for a higher frequency of truck arrivals. But it should remain low since there are few pallets in the trucks and they spend most of their time in the warehouses.

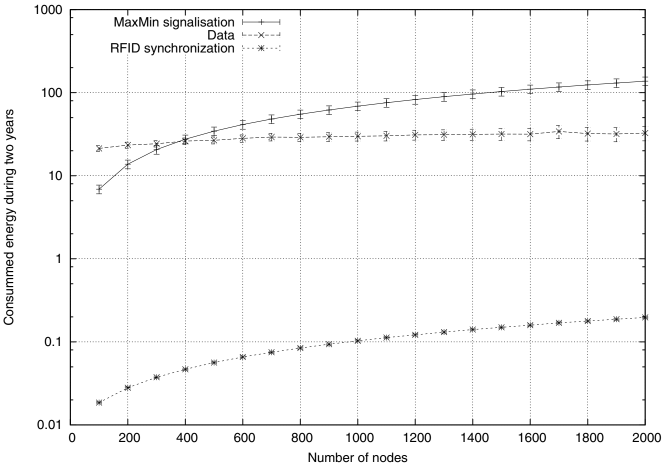

Figure 3: energy consumption (in Joules) for the different traffic types in function of the number of nodes in the networks

B. *Estimation of the synchronization delays in the truck*

It is interesting to have an idea of the time needed for the sensors to synchronize. In the warehouses, the delays is smaller than in the truck where the density of new sensors put together is higher than in the warehouses. Let $E[T_1]$ represent the average waiting time for the first sensor to synchronize when it arrives into a truck. $E[T_i]$ is the average response time (waiting plus service times) for the i-*th* node to synchronize.

$$E[T_1]=C+E[\tau_1]+T_{request}+N_{respons}*T+T_{Ack}+\theta_{margin\ slots}$$

$$E[T_i]=E[T_{i-1}]+E[\tau_i]+T_{request}+N_{respons}*T+T_{Ack}+\theta_{margin\ slots}$$

Where $i\in[1,\ldots,N_{Truck}]$, N_{Truck} is the total number of sensors in the truck, C is the time duration of one period (duty-cyle), $N_{respons}$ is equal to N_{Truck} , $T_{request}$ and T_{Ack} are taken equal to T (4.25ms), $\theta_{margin\ slots}$ is equal to 8*T. The nodes wait a random variable τ_i uniformly distributed over [0; T_{max}], this time indicating when to trigger their RFID. The average $E[\tau_i]$ is thus T_{max}/N_{Truck} for all i, where T_{max} is 20minutes. The delay to synchronize in the truck is upper bounded by 2 cycles of 20minutes each. It is interesting to notice that this delay decreases with the number of nodes. Actually, $E[\tau_i]$ which is dominant in the formula decreases with the number of nodes but the global synchronization time remains almost the same.

As a result the delays estimated by the proposed models are compatible with the time spent in the warehouse and the transportation time inside the truck.

V. CONCLUSION

In this paper we propose a self-organization mechanism for an autonomous end to end cold chain monitoring system. We argue that no global mechanism is optimal in the case of the cold chain but a mechanism should use the protocols which are optimized for the different part of this chain and dynamically switch between them depending on the context. It is crucial the system require no infrastructure because the logistic actors are too preoccupied with the increase of the price of the petroleum and logistic transportation specific costs to be ready to accept an additional expense. Our proposed CAM does not require any infrastructure: it autonomously detects in which context a node is and it requests to apply the best protocols accordingly. We introduced the concept of dynamic synchronization method combined with a dynamic choice of routing algorithms (PLACIDE/MAXMIN). To easily manage the mobility and also for energy consumption minimization, we use wake up signals to activate a sleeping sensor. We evaluated the performance of our proposed mechanism. The results show that the consumed energy remains acceptable and the synchronization delays too.

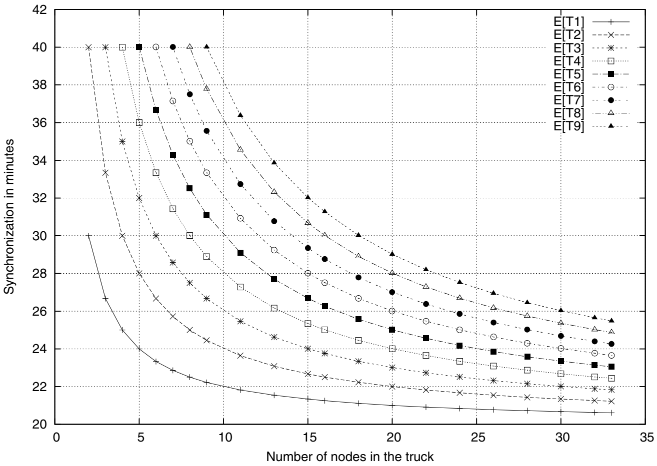

Figure 4: synchronization time in the truck in function of the number of nodes in the truck

VI. ACKNOWLEDGEMENT

The authors want to thank Mr. Vincent Gauthier for useful comments on this paper.

VII. REFERENCE

[1] C. Nelson, P. Froes, A. M. Van Dyck, J. Chavarria, E. Boda, A. Coca, G. Crespo, H. Lima, "Monitoring temperatures in the vaccine cold chain in Bolivia", Vaccine, Volume 25, Issue 3, 5 January 2007.

[2] R. Riem-Vis, “Cold chain management using an ultra low power wireless sensor network”, Workshop on Applications of Mobile Embedded Systems. Boston, 2004.

[3] W. Fu, Y. S. Chang, M. M. Aung, C. Makatsoris, C. H. Oh, “WSN based intelligent cold chain management”, The 6th ICMR’08, Brunel University, UK, 9-11th September, 2008.

[4] R. Kacimi, R. Dhaou, A. Beylot, “Using Energy-Efficient Wireless Sensor Network for Cold Chain Monitoring”, IEEE CCNC, 2009, USA.

[5] I. Rhee, A. Warrier , M. Aia , J. Min, “Z-MAC: a hybrid MAC for wireless sensor networks”, Proceedings of the 3rd international conference on Embedded networked sensor systems,2005 USA

[6] A. Amis, R. Prakash, Thai. P, V. Dung ,T. Huynh, “Max-Min D-Cluster Formation in Wireless Ad Hoc Networks” ,in (2000) IEEE INFOCOM.

[7] A. Delye, M. Marot, M. Becker, "Correction, Generalization and Validation of the Max-Min d-cluster formation heuristic", TC6/LNCS NETWORKING 2007

[8] R. Jurdak, A. G. Ruzzelli, G. M. P. O’Hare, “Multi-hop RFID Wake-up Radio: Design,Evaluation and Energy Tradeoff”, (ICCCN), 2008.

[9] L. Gu, J.A. Stankovic,"Radio-triggered wake-up capability for sensor networks," 10th IEEE Real-Time and Embedded Technology and Applications Symposium, 2004. pp. 27- 36, 25-28 May 2004

[10] I. Rhee, A. Warrier, M. Jeongki, X. Lisong, "DRAND: Distributed Randomized TDMA Scheduling for Wireless Ad Hoc Networks," *Mobile Computing, IEEE Transactions on* , vol.8, no.10, Oct. 2009