Technical Report

Real-Time Tracking and Reporting of Dynamic Events in Hierarchical Wireless Sensor Networks

Anis Koubâa

Hassen Sallay

Mário Alves

HURRAY-TR-070503

Version: 1.0

Date: 03/MAY/2007

Real-Time Tracking and Reporting of Dynamic Events in Hierarchical Wireless Sensor Networks

Anis Koubaa[1,2], Hassen Sallay[2], Mário Alves[1]

[1]IPP-HURRAY! Research Group

Polytechnic Institute of Porto – School of Engineering

Rua Dr. António Bernardino de Almeida, 431, 4200-072 Porto, Portugal

akoubaa@dei.isep.ipp.pt, mjf@isep.ipp.pt

[2]Al-Imam Muhammad Ibn Saud University

College of Computer Science and Information Systems

11681 Riyadh, SAUDI ARABIA

hsallay@ccis.imamu.edu.sa

Abstract

Wireless Sensor Networks (WSNs) are highly distributed systems in which resource allocation (bandwidth, memory) must be performed efficiently to provide a minimum acceptable Quality of Service (QoS) to the regions where critical events occur. In fact, if resources are statically assigned independently from the location and instant of the events, these resources will definitely be misused. In other words, it is more efficient to dynamically grant more resources to sensor nodes affected by critical events, thus providing better network resource management and reducing end-to-nd delays of event notification and tracking. In this paper, we discuss the use of a WSN management architecture based on the active network management paradigm to provide the real-time tracking and reporting of dynamic events while ensuring efficient resource utilization. The active network management paradigm allows packets to transport not only data, but also program scripts that will be executed in the nodes to dynamically modify the operation of the network. This presumes the use of a runtime execution environment (middleware) in each node to interpret the script. We consider hierarchical (e.g. cluster-tree, two-tiered architecture) WSN topologies since they have been used to improve the timing performance of WSNs as they support deterministic medium access control protocols.

Document history

Version	Content	By	Date
1.0		AK, HS, MA	03/MAY/07

Real-Time Tracking and Reporting of Dynamic Events in Hierarchical Wireless Sensor Networks

Anis Koubâa[1,2], Hassen Sallay[2], Mário Alves[1]

[1] IPP-HURRAY! Research Group, Polytechnic Institute of Porto
Rua Dr. Antonio Bernardino de Almeida, 431, 4200-072 Porto, PORTUGAL

[2] Al-Imam Muhammad Ibn Saud University, College of Computer Science and Information Systems, 11681 Riyadh, SAUDI ARABIA

akoubaa@dei.isep.ipp.pt, hsallay@ccis.imamu.edu.sa, mjf@isep.ipp.pt

Abstract

Wireless Sensor Networks (WSNs) are highly distributed systems in which resource allocation (bandwidth, memory) must be performed efficiently to provide a minimum acceptable Quality of Service (QoS) to the regions where critical events occur. In fact, if resources are statically assigned independently from the location and instant of the events, these resources will definitely be misused. In other words, it is more efficient to dynamically grant more resources to sensor nodes affected by critical events, thus providing better network resource management and reducing end-to-end delays of event notification and tracking. In this paper, we discuss the use of a WSN management architecture based on the active network management paradigm to provide the real-time tracking and reporting of dynamic events while ensuring efficient resource utilization. The active network management paradigm allows packets to transport not only data, but also program scripts that will be executed in the nodes to dynamically modify the operation of the network. This presumes the use of a runtime execution environment (middleware) in each node to interpret the script. We consider hierarchical (e.g. cluster-tree, two-tiered architecture) WSN topologies since they have been used to improve the timing performance of WSNs as they support deterministic medium access control protocols.

1. Introduction

Providing real-time guarantees in Wireless Sensor Networks (WSNs) has been considered as one of the most challenging topics due to the large-scale nature of these networks and to the severe physical constraints inherent to sensor nodes [1]. Several research works focusing on real-time support in WSNs have concentrated their investigation on using hierarchical WSNs topologies to provide time guarantees [2-6]. In this paper, the *hierarchical WSN model* refers to the structured cluster-based WSN topology. In fact, in contrast to the flat peer-to-peer communication model, which usually relies on contention-based mechanism (e.g. CSMA/CA) as Medium Access Control (MAC) protocols, hierarchical cluster-based topologies have been shown to be quite suitable for WSNs with demanding requirements in terms of Quality of Service (QoS) requirements, namely timeliness [2]. Real-time communications in hierarchical WSNs is mainly achieved by either using deterministic MAC protocols such as Time Division Multiple Access (TDMA) based protocols [5] or two-tiered architectures [4, 7, 8]. Basically, hierarchical WSNs are divided into a number of inter-connected clusters where each cluster is managed by a central node referred to as *router* (or cluster-head). Nodes inside a given cluster send their packets to their *parent* router which relays them to the destination through the backbone of routers, which might be structured as a tree [2], or hexagon [3].

In WSN applications, events can be either *static* or *dynamic*. Static events occur in a specific location and do not move in time (e.g. monitoring temperature, humidity). These events are easy to predict and to measure. In contrast, dynamic events are typically mobile when they are evolving during time (e.g. target tracking, intrusion detection, fire detection, etc.), which induces an additional complexity to report them and track their evolution in space and time. While static events can be managed using static resource allocations since they are easy to predict, an efficient detection and tracking of dynamic events requires a dynamic allocation of network resource that depends on the spatiotemporal evolution of the events.

In hierarchical WSNs, resource allocation is basically related to the amount of bandwidth granted for each router to deliver data. Roughly, the allocated bandwidth in hierarchical TDMA-like networks is typically measured by the amount of time a certain node is allowed to inject traffic in the network at a certain data rate. In the literature, bandwidth allocation is mainly performed statically by taking into account some parameters of the network (e.g. energy, depth of the network, number of child routers per parent router, etc.), but it does not dynamically adapt to the location and the time where and when the events occur. For instance, LEACH – proposed in Reference [5] - is an adaptive clustering protocol that dynamically changes the cluster-heads based on their energy levels in each round, but it supports a static TDMA schedule during each round independently of the occurrence of any event. The dynamic behavior of this protocol is only related to the energy level, but not to the dynamics of the events which limits its use for *tracking* and *reporting* the information of dynamic events in real-time. In this paper, *event tracking* means the monitoring of its evolution during time and *event reporting* means the delivery of messages describing the event to the control station. In addition, in Reference [2] the authors have proposed a model for the worst-case dimensioning of cluster-tree networks (instantiated to the IEEE 802.15.4/Zigbee protocols [9, 10]) using a static bandwidth allocation in routers that typically depends on the depth of the tree and the number of child routers per parent router. Although the model provides an efficient way to estimate the delay bounds in such networks, it does not take into consideration the dynamic nature of the evolved events, which limits its efficient applicability to the detection of static events. It results that the bandwidth must be allocated in an efficient way when considering dynamic events (e.g. target tracking) in hierarchical WSNs.

Performing dynamic resource allocation in hierarchical WSNs is a challenging issue that requires the deployment of distributed algorithms, which trigger a cooperative work between the different sensor nodes to achieve a predefined task. However, distributed algorithms in WSNs may be the origin of unnecessary communication overheads if they are not implemented efficiently. For instance, it is more efficient to activate tracking-related services/policies only in the sensor nodes involved in the tracking of an event, rather than in the entire network. In addition, delegating some specific management tasks to some sensor nodes in the network, i.e. in-network dynamic processing, will be of a great help to reduce the communication overheads and thus saving energy, since it is not necessary to report raw data directly to the destination. It is therefore necessary to provide a run-time environment, i.e. middleware that facilitates the deployment of distributed algorithms, particularly those ensuring the tracking and the reporting of evolving events, and also that enables the dynamic configuration of the network, which ensures the efficient management of network resources and the real-time reporting and tracking of the events. Those requirements can be achieved by using the *active network management paradigm*, which has been considered in the literature as a quite suitable approach for handling the dynamicity of networks. In the active network approach, packets are allowed to transport processing codes (or scripts) that will be executed in the routers/nodes by a special run-time environment in the intermediate nodes. The idea behind active network management is to improve the flexibility of programming the sensor network and changing its behavior dynamically.

In this paper, we propose, *Activ-WiSe*, a management architecture based on the active network paradigm for supporting real-time tracking and reporting of dynamic event in hierarchical WSNs. We also show a strategy for the deployment of the active network paradigm for efficient tracking and reporting of dynamic events.

2. Related Work

Dynamic resource management in WSNs has been addressed in several research works, from different perspectives.

In the literature, the active network management paradigm has been used as a key technique for controlling the dynamics of sensor networks (e.g. [11-13]). Initially, the Active Network paradigm has been proposed for the Internet [14], where packets are allowed to transport scripts that will be executed by a runtime environment (also referred to as middleware) in the routers to dynamically modify the operation of the network or to dynamically deploy new software for activating new services during network runtime. This communication paradigm has been extended to support dynamic service deployment in WSNs, which enables to perform the dynamic re-configuration of the network. For instance, in Reference [11], the authors have proposed a framework for programmable sensor networks, called *SensorWare*. In that paper, it has been shown that the use of an active network framework based on a lightweight scripting approach can improve the management in WSNs by allowing the online programming of sensor nodes and by providing an efficient infrastructure for running task-specific distributed algorithms. In Reference [13], Levis et al. have made an abstraction of the programming models and have proposed a general architecture for implementing the underlying middleware corresponding to a programming model. The architecture is based on a virtual machine template composed of (1) *handlers*, which are code routines that run in response to an operating system event (e.g. timers, route forwarding request, etc), (2) *capsules*, which are the units of code that propagates throughout the network, and (3) *operations*, which are user-defined actions that will executed when an event is detected. In Reference [12], the authors have used the active network paradigm as a means to implement a decentralized architecture for gathering distributed information.

Several other works have also investigated the use of the active network communication models for different purposes, but with the common objective of dynamic deployment of distributed algorithms. Those works, however, were designed independently of the WSN topology and were not optimized for supporting real-time guarantees, which is the main target of our work. Our contribution in this paper resides in taking advantage of the active network management communication model to improve the responsiveness of event tracking and reporting in hierarchical WSNs.

Our objective is to design an abstract model of an active network architecture specifically for hierarchical WSNs to (1) push the processing as close as the evolved events (2) enable dynamic bandwidth allocation in all routers involved into the tracking and reporting of the events, and (3) improve the timing guarantees in hierarchical WSNs.

To summarize, the contributions of this paper are two folded:

1- Design of an abstract active network model for hierarchical WSNs,
2- Proposal of a methodology for dynamic bandwidth resource management in WSNs,
3- Proposal of a strategy for the deployment of the active network paradigm for efficient tracking.

The rest of this paper is organized as follows. Section 3 describes the *Activ-WiSe* architecture. Section 4 describes the methodology for efficient and dynamic bandwidth allocation. It also presents the strategy for the deployment of the active network paradigm for efficient tracking. Section 5 concludes the paper.

3. Activ-WiSe: Active Network Management Architecture for Hierarchical WSNs

In this section, we present an abstract management architecture for hierarchical WSNs. The proposed architecture provides a two-level hierarchy: the *root level*, and the *cluster node level*. At each level of the hierarchy a dedicated management agent is operational (see Fig. 1).

At the root level, a *Root Agent* (RA) is in charge of the management activities. The RA hosts a data storage facility that has the entire data related to each specific service. At the cluster node level, *Node Agents* (NA) are deployed. The deployment or the activation of these agents is dynamically performed according to the characteristics of the managed service (e.g. tracking, security, performance measurements, etc.). These node agents interact with the sensor nodes within their clusters in order to collect some data related to a specific service (e.g. number of sensor nodes per cluster as input for the network coverage service). Each node agent has a local view of its cluster, maintains a relation with the other node agents involved in the same service, as well as with the RA. This model is more scalable than a full root-driven polling approach. Moreover, it enables the root agent to build service level statistics that are specific to the service level management process in use for the service (e.g. checking the conformity of the delivered service to the desired level).

Choosing active network technology to implement our architecture allows us to exploit its benefit in terms of flexibility and improved scalability. These benefits are mainly the following:

- **Dynamic service deployment.** It is needed in hierarchical WSNs to perform a dynamic deployment of services in the sensor nodes. In fact, a WSN deploys - in general - different services related to the network management such as fault-tolerance, security, configuration, etc. Setting-up and activating all the services in all routers of the network is not efficient in terms of storage, processing and memory management in sensor nodes, due to their reduced processing and storage capabilities. It is more suitable to perform *on demand* set-up and activation of the required services in sensor nodes; In fact, at a given time we may want to set-up and/or activate a particular service only in a certain number of routers that will cooperatively perform a specific task. For instance, in a target tracking application, a given event may trigger the execution of a set security measures according to event-specific policies only on the nodes concerned by the event.

- **Improvement of the communication efficiency through in-network processing.** In highly distributed and large-scale sensor network applications, it is quite suitable to optimize the communication by delegating some management and processing functions (e.g. aggregation, data collection nodes) to specific routers inside the network instead of having a centralized processing in the root. In this case, when an event occurs the messages will be routed to a certain parent router that has been delegated by the root to perform a particular processing or management task, which will simply transfer periodic or on demand reports to the root.

We adopt the active network approach to dynamically download and execute node agents from the repository located in the root agent. It is also possible to request and download the service from the nearest parent (towards the root) involved in this service. The availability of a dynamic code distribution facility and the presence of execution environments (e.g. middleware) on all nodes enable to easily and dynamically add new management functions in particular nodes, i.e. to push management functions where they are needed. We call our architecture *Activ-WiSe*, Active-based Management Architecture for Wireless Sensor Networks.

The Activ-WiSe architecture is illustrated in Fig. 1. Within Activ-WiSe, WSN management functions are designed as a set of dedicated active applications called *plug-ins*. Our architecture can be integrated with any existing active execution environment such as those proposed in References [11, 13]. These plug-ins are stored in a repository located in the root agent.

Once installed on the root agent, these plug-ins can be downloaded by node agents where they will be executed. The plug-ins uninstall themselves when they no longer be used. The Activ-WiSe architecture basically considers the traditional FCAPS management areas, which refer to **F**ault and **C**onfiguration management, **A**ccounting and **P**erformance management and finally **S**ecurity management. Each area is represented by a specific plug-in performing the corresponding management task.

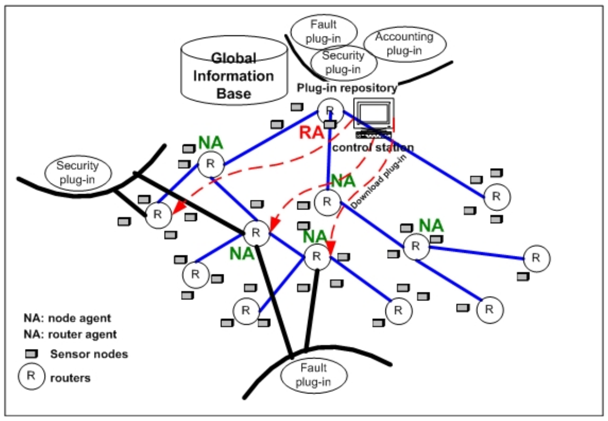

Fig. 1. The Activ-WiSe architecture

In addition to the plug-ins, a management data exchange protocol is assumed to enable the communication between the different plug-ins either with the plug-ins in the same node agent or with those in other node agents.

4. Efficient Resource Management in Hierarchical Wireless Sensor Networks

In this section, we propose a methodology for assigning the adequate bandwidth to each router in hierarchical WSNs for sake of ensuring an efficient tracking and reporting of dynamic events. We also demonstrate how to use the active network approach to delegate specific tasks to certain routers to optimize the use of resources in the network.

4.1. Problem statement

Static bandwidth allocation for hierarchical WSNs, such as those proposed in References [2, 3], is definitely not efficient for the tracking and the reporting of dynamic events. In fact, it is more effective that routers involved in the monitoring of the dynamic event are granted bandwidth such that the responsiveness of the event reporting is reduced, i.e. the timing requirements are satisfied.

Network model and assumptions; We assume a hierarchical WSN composed of different routers, where each router is the head of a certain cluster. Each cluster contains sensor nodes associated to the cluster-head router. We assume that the hierarchical network provides a complete coverage of the monitored region. In this study, we consider the tracking and the reporting a *single* event. Multiple events monitoring is more challenging and is out of the scope of this paper. We also assume that the sensory traffic is routed to a monitoring station attached to a particular router that has a complete knowledge of the network structure (such as the *root* in cluster-tree networks [2] as shown in Fig. 2). Initially, the network may be assumed to operate with a specific bandwidth allocation independent of any event. We assume that the network is based on a TDMA-like MAC protocol where each router is granted a specific time window to send his data. The length of this time window is **proportional** to the amount of bandwidth granted for its up-stream link. For instance, this model has been adopted by the IEEE 802.15.4/Zigbee cluster-tree networks [15].

Let us consider an event *E* moving inside the monitored region. This event is first detected by the sensor nodes close to it, which send their reports to their respective cluster-head. Each cluster-head must then forward this report to the remote control station, which can be the root and/or any other router in the network. In this paper, we have two problems (1) the first problem is how to efficiently assign bandwidth resources in the routers to achieve the best performance in terms of real-time monitoring of the event. (2) The second problem is how to deploy our plug-ins in the WSN according to the service characteristics in order to achieve the optimal distribution of the plug-ins minimizing the management overhead of the reporting and tracking processes.

Roughly, the problem can be formulated as follows.

Given a hierarchical WSN and a detected event, what are the best distribution of bandwidth among all the routers and the optimal distribution of the plug-ins among WSN that improves the responsiveness tracking and reporting of this event and optimizes the usage of resources?

4.2. A monitoring methodology of dynamic events

To give an intuition on the general methodology, let us consider a hierarchical network such as shown in Fig. 2. When the events are known in advance (periodic polling of sensor nodes), bandwidth and memory resources may be statically assigned to routers according to a certain model (e.g. [2]). However, assuming that the occurrence of an event cannot be predicted in space and time, it is more efficient to adapt the bandwidth and memory allocation such that routers contributing to the delivery of the evolved event *E* to the destination will be granted higher resources (thick lines with arrows in Fig. 2). For instance, when event *E* occurs, the paths $R5 \rightarrow R2 \rightarrow Root$ and $R9 \rightarrow R6 \rightarrow R2 \rightarrow Root$ must be granted higher amount of bandwidth than the remaining routers in the network. In what follows, we refer to a cluster *i* by the name of its router R_i.

When the event *E* occurs, the clusters R_5 and R_9 will detect this event and will report it to the monitoring station attached to the root. Two interesting issues can be initiated by the root after the reception of the first messages describing event *E*:

1. The delegation of the tracking task to the router R_2 since it is a common parent (not necessarily direct parent) to both routers R_5 and R_9. This delegation results in performing the in-network processing of the raw data rather sending it directly to the monitor. This fact will help to improve the bandwidth resource usage on the link $R2 \rightarrow Root$. It also results in reducing the energy consumption since there is no need to send individual raw data messages on the link $R2 \rightarrow Root$.
2. The re-allocation of the bandwidth resources such that the communication links involved in the reporting of the event (i.e. $R9 \rightarrow R6 \rightarrow R2 \rightarrow Root$ and $R5 \rightarrow R2 \rightarrow Root$) will be granted higher bandwidth to improve the timing performance of the delivery.

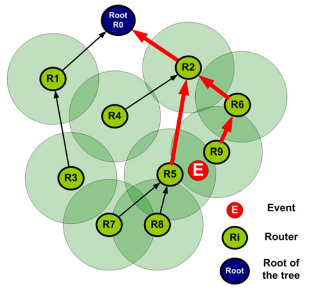

Fig. 2. Illustrative example of dynamic allocation

Finally, after computing the new assignment of the bandwidth on the different links, it is necessary to re-synchronize the network according to the new bandwidth distribution, since the bandwidth is proportional to the activity window length of the TDMA-like schedule. In other words, each router must be allocated a collision-free time window proportional to the amount of bandwidth that it has been assigned.

To summarize, the monitoring methodology of dynamic events comprises four steps:

1. *Detection of the event*. The event is detected by the sensor nodes close to it, which will initially send the report to remote monitoring station (i.e. the root).
2. *Task delegation*. When the root detects that a given event is reported from different sensor nodes, it will determine the closest common parent and delegate to that parent the task of gathering and processing the raw data of the event before forwarding them to the root. We refer to this special node as *delegate router*. It in this case, it is assumed that the root has knowledge of the parent-to-child relationship between nodes. This assumption is realistic, since for instance the Zigbee standard has adopted an addressing scheme that enables to determine the depth and the location of a node in the network [10]. The task delegation is supported by the active network middleware that will activate/set-up the adequate service to perform this operation. This in-network processing will help on reducing energy consumption and on improving the tracking quality of the event.
3. *Bandwidth assignment*. The root must compute the new TDMA schedule that provides more activity periods (or greater time slots), thus more bandwidth, for sensor nodes involved in the tracking of the event. In the next section, we discuss different strategies for assigning adequate bandwidth resources in the network.
4. *Re-synchronization*. After the computation of the new TDMA schedule, the network must re-

synchronize itself accordingly. This requires the deployment of distributed algorithms for re-synchronizing the network. In the literature, several works have focused on distributed synchronization in TDMA-like networks, such as for instance [16, 17]. The re-synchronization of the hierarchical network is out of the scope of this paper and will be considered in future works addressing cluster-tree WSNs.

4.3. Bandwidth assignment in hierarchical WSNs

In this section, we propose some strategies to perform efficient bandwidth distribution in hierarchical WSNs.

When tracking an event E, we can basically distinguish three types of clusters as adopted in Reference [18]:

1- *Active clusters*: they are the clusters that currently (directly or indirectly) detect the event E (e.g. clusters R_5, R_9, R_6 and R_2 in Fig. 2). Thus, the cluster-heads of these clusters must be granted the maximum bandwidth along the path to the root or to the delegate router. Certainly, the bandwidth assigned may be different from one active cluster to another depending on some characteristics of the network (e.g. depth in cluster-tree WSNs).

2- *Inactive (or sleeping) clusters*: they are the clusters that are not involved in the tracking of the event. As a consequence, these clusters require an amount of bandwidth smaller than that in active clusters.

3- *Alerted clusters*: they are clusters that have not yet detected the event, but expect it to enter the cluster soon. An inactive cluster may become in an alerted state according to some particular strategies. For instance, if the cluster-head of an inactive cluster receives a message about the target detection from a neighbor cluster-head of an active cluster.

In order to adequately assign the bandwidth to each router, we must establish the set of equations that express the constraints on the bandwidth of each router.

Let us define the *duty cycle* (DC) of a given cluster as the ratio of its time slot duration divided by its period, i.e. the time between two consecutive time slots. It can be easily observed that the output bandwidth is proportional to the duty cycle of the cluster. In the following, we derive three general conditions that must be satisfied by the duty cycle.

- **First constraint.** The sum of all bandwidths must be lower or equal to the maximum bandwidth of the network. Hence, we consider the following equation expressing this constraint, assuming that there are N routers in the network:

$$\sum_{i=1}^{N} DC_i \leq 1 \qquad \textbf{(1)}$$

- **Second constraint.** The *minimum* duty cycle of an active cluster $DC_{\min}^{active}$ must be greater than the *minimum* duty cycle of an alerted cluster $DC_{\min}^{alerted}$, which is in turn greater than the *minimum* duty cycle of an inactive cluster $DC_{\min}^{inactive}$. We propose the following generic equation expressing the second constraint:

$$DC_{\min}^{active} = \alpha \cdot DC_{\min}^{alerted} = \beta \cdot DC_{\min}^{inactive} \qquad \textbf{(2)}$$

where $\alpha > \beta$ are application-specific constant coefficients greater than 1.

Note that the *minimum duty cycle of an inactive cluster* is equally assigned to all routers in an inactive state. Reference [15] proposed a different approach such that the minimum duty cycle in a cluster is assigned to the leaf cluster-heads in the hierarchical network (e.g. the routers with the highest depth in the cluster-tree topology) since they are the furthest from the monitoring station (i.e. the root) and are less involved in the routing process. In our case, we do not care about a topology-aware distribution of duty cycles in inactive clusters since we only focus on improving the tracking and the reporting in active clusters. The *minimum duty cycle of an active cluster* is assigned to the active cluster-heads that have detected the event in their clusters and are not involved in the routing process. The duty cycle assignment for other active clusters will be described in the third constraint. Finally, the *minimum duty cycle of an alerted cluster* is equally assigned to all alerted clusters.

- **Third constraint.** The duty cycle of an *active* parent router must be equal to the sum of all duty cycles of its *active* child routers. This is because the activity period of the *active* parent router must be able to support all the ingoing traffic from its *active* children.

$$DC_{parent}^{active} - \sum DC_{children}^{active} = 0 \qquad \textbf{(3)}$$

Thus, it is possible to derive a set of linear equation that can be easily resolved using the theory of linear algebra.

For instance, let us consider the example in Fig. 2. Applying the three constraints, it is possible to write the following linear system, assuming that R_8 is an alerted cluster:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -\beta & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -\alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\beta & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\beta & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\beta & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} DC_0 \\ DC_1 \\ DC_2 \\ DC_3 \\ DC_4 \\ DC_5 \\ DC_6 \\ DC_7 \\ DC_8 \\ DC_9 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad \textbf{(4)}$$

Observe that the first line in the Matrix corresponds to Eq. (1), the lines 2, 5, 6 and 7 correspond to the third constraint in Eq. (3) and the remaining lines correspond to the second constraint. We can show that such a system comprises only one solution since the number of equations is at least equal to the number of unknown variables. It is also possible to resolve the linear equation system even if the system is not square. Computational tools such as MATLAB easily process such an equation system.

From a practical point of view, this computation of the new bandwidth assignment cannot be in the sensor nodes (even the root) due to their limited computational power. However, this can be performed by the monitoring machine attached to root or any other router and then distribute the new assignment in the network.

When the event moves, it may be necessary to change the bandwidth assignment as the event enters a new cluster which will be upgraded to active cluster. The active network paradigm helps on the management of these dynamic changes of the network configuration, as well as in the re-synchronization process.

4.4. Plug-ins deployment strategy in hierarchical WSNs

The dynamicity of the events can be viewed from two levels: (1) the *micro-dynamic* level and (2) the *macro-dynamic* one. Through the micro-dynamic level, the local evolution of the event within a single cluster node can be sensed and monitored. We designate this process by ***local tracking process***. The macro-dynamic level represents the evolution of the event globally through the cluster-tree network. We designate this process by the ***global tracking process***. By differentiating these two levels and designing for each tracking process a specific plug-in, the overhead generated by the tracking and reporting tasks can be reduced. Only data that is mandatory for a tracking task is sent over the network. In the rest of this section, we propose a strategy performing an efficient deployment of these plug-ins among an hierarchical WSNs.

The strategy takes as input parameter a set of active clusters i.e. the clusters that currently detect the event E (e.g. clusters R_5, R_9, R_6 and R_2 in Fig. 2) called ***local tracking candidates set (CS)*** and return as output three parameters: (1) the node where the plug-in must be deployed for performing local tracking process (called ***local tracking elected node (LEN)***). The LEN necessarily belongs to the CS set, (2) the node where we must place the plug-in performing the global tracking process (called ***global tracking elected node (GEN)***) (3) a set of alerted clusters ***(AC)*** i.e. clusters that have not yet detected the event, but expect it to enter the cluster soon.

The strategy applies the following rules:

1- The *global tracking elected node* is the first common parent of the nodes belonging to the *tracking candidates set*,

2- The *local tracking elected node* is the nearest cluster–head to the sensed event, i.e. that has the best estimate of the event of its position.

3- The *alerted cluster set* will become the *tracking candidates set* except LEN union the set of all the CS childes

Rule1 eliminates the reporting overhead between the GEN and the cluster-tree root. *Rule 2* ensures the highest degree of the event localization accuracy. *Rule 3* eliminates the reporting redundancy by delegating the reporting task to a unique cluster-head (LEN) instead of all the CS nodes.

By applying the above strategy, the overhead generated by the tracking and reporting tasks is significantly reduced.

5. Concluding Remarks

In this paper, we have presented a methodology for the tracking and the reporting of dynamic events. We have proposed the Activ-WiSe management architecture, which is of an active network framework designed for hierarchical WSNs to manage the dynamic changes in the networks, namely in terms of bandwidth management. We have also proposed an efficient bandwidth resource management strategy for sake of ensuring an efficient tracking and reporting of dynamic events. In fact, the proposed methodology is shown to reduce the responsiveness of the event reporting and to provide fair share of bandwidth resources. A strategy for the deployment of active plug-in for efficient tracking has also been proposed.

In future works, we intend to elaborate more on the Activ-WiSe architecture and its deployment for a real-time target tracking and reporting application. We will particularly address the problem network re-synchronization by the proposal of distributed algorithms that assign the new bandwidth allocation to the routers. Furthermore, We are currently looking forwards to integrate Activ-WiSe management architecture in the context of ART–WiSe architecture [4]. An implementation of Activ-WiSe is also envisaged.

References

[1] J. A. Stankovic, T. Abdelzaher, C. Lu, L. Sha, and J. Hou, "Real-Time Communication and Coordination in Embedded Sensor Networks," *Proceedings of the IEEE*, vol. 91, pp. 1002-1022, 2003.

[2] A. Koubaa, M. Alvès, and E. Tovar, "Modeling and Worst-Case Dimensioning of Cluster-Tree Wireless Sensor Networks," in *27th IEEE Real-Time Systems Symposium (RTSS'06)*. Rio de Janeiro (Brazil), 2006.

[3] S. Prabh and T. Abdelzaher, "On Scheduling and Real-Time Capacity of Hexagonal Wireless Sensor Networks," in *Euromicro Conference on Real-Time Systems*. Pisa (Italy), 2007

[4] A. Koubâa, M. Alves, and E. Tovar, "A Two-Tiered Architecture for Real-Time Communications in Large-Scale Wireless Sensor Networks: Research Challenges," in *17th Euromicro Conference on Real-Time Systems (ECRTS'05), WiP Session,*. Palma de Mallorca (Spain), 2005.

[5] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocols for wireless microsensor networks," in Proceedings of the Hawaii International Conference on Systems Sciences, Hawai, 2000.

[6] S. Subramaniam, V. Kalogeraki, and T. Palpanas, "Distributed Real-Time Detection and Tracking of Homogeneous Regions in Sensor Networks " in *27th IEEE International Real-Time Systems Symposium (RTSS'06)*. Rio de Janeiro (Brazil): IEEE, 2006.

[7] V. A. Kottapalli, A. S. Kiremidjiana, J. P. Lyncha, E. Carryerb, T. W. Kennyb, K. H. Law, and Y. Lei, "Two-Tiered Wireless Sensor Network Architecture for Structural Monitoring," in Proceedings of the 10th Annual International Symposium on Smart Structures and Materials, San Diego (USA), 2003.

[8] G. Gupta and M. Younis, "Fault-Tolerant Clustering of Wireless Sensor Networks," in *IEEE Wireless Communication and Networks Conference (WCNC 2003)*, vol. 3. New Orleans (Louisiana), 2003.

[9] IEEE-TG15.4, "Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)," *IEEE standard for Information Technology*, 2003.

[10] Zigbee-Alliance, "ZigBee specification," *http://www.zigbee.org/*, 2005.

[11] A. Boulis and M. B. Srivastava, "A framework for efficient and programmable sensor networks," in *Proceedings of IEEE Open Architectures and Network Programming (OPENARCH)*. Los Angeles (USA): IEEE, 2002.

[12] A. Makarenko, A. Brooks, S. Williams, H. Durrant-Whyte, and B. Grocholsky, "A Decentralized Architecture for Active Sensor Networks," in *IEEE International Conference on Robotics and Automation (ICRA'04)*. New Orleans (USA): IEEE, 2004.

[13] P. Levis, D. Gay, and D. Culler, "Active Sensor Networks," in *The Second USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI 2005)*. Boston (USA), 2005.

[14] D. L. Tennenhouse and D. J. Wetherall, "Towards an Active Network Architecture," *Computer Communication Review*, 1996.

[15] A. Koubâa, A. Cunha, and M. Alves, "A Time Division Beacon Scheduling Mechanism for IEEE 802.15.4/Zigbee Cluster-Tree Wireless Sensor Networks," in *Euromicro Conference on Real-Time Systems (ECRTS 2007)*. Pisa(Italy), 2007.

[16] M. Caccamo and L. Y. Zhang, "The Capacity of an Implicit Prioritized Access Protocol in Wireless Sensor Networks," *Journal of Embedded Computing (JEC) by Cambridge International Science Publishing*, vol. 1, 2005.

[17] T. L. Crenshaw, A. Tirumala, S. Hoke, and M. Caccamo, "A Robust Implicit Access Protocol for Real-Time Wireless Collaboration," in *IEEE Euromicro Conference on Real-Time Systems*. Palma de Mallorca (SPAIN), 2005.

[18] R. R. Brooks, P. Ramanathan, and A. M. Sayeed, "Distributed Target Classification and Tracking in Sensor Networks," *Proceedings of the IEEE*, vol. 91, pp. 1163 - 1171, 2003.