Analyzing The Performance Of A Self Organizing Framework For Wireless Sensor-Actuator Networks

Muhammad Farukh Munir and Fethi Filali
Institut Eurécom, Department of Mobile Communications,
Sophia Antipolis, France
Email: {munir, filali}@eurecom.fr

Keywords: coordination, self organization, routing, mobility, delay-energy aware protocols.

Abstract

Wireless sensor-actuator networks (WSANs) refer to a group of sensors and actuators linked by a wireless medium to perform distributed sensing and acting tasks.

In order to provide effective sensing and acting, coordination mechanisms are required among sensors and actuators. We have presented the architectural design of a self organizing coordination framework in [3]. The framework organizes the two-tiered heterogeneous network into clusters. The coordination framework is a three-level coordination protocol and the three levels belong to the three types of coordination required for WSANs: sensor-sensor coordination, sensor-actuator coordination and actuator-actuator coordination. In WSANs (or WSNs), the routing protocol is highly influenced by the operating application with delay and energy consumption constraints. Therefore, we propose to decouple the application dependent coordination dynamics from the essentials provided by the routing protocol.

In this paper, we present the performance analysis of our proposal called ADP (Actuator Discovery Protocol) for static and mobile WSANs. Using ns-2 simulation results we have shown to eliminate redundancy at each coordination level to make routing effective and fault-tolerant.

1. INTRODUCTION

The future of WSNs is to embed numerous distributed devices to monitor and interact with physical world phenomena, and to exploit their spatially and temporally dense sensing and actuation capabilities. Recent advances in technology have lead to the emergence of SANETs[1] giving a distributed control to the management, communication and coordination aspects of the network functioning formerly referred to as WSNs. As depicted in Figure 1, a SANET consists of a group of sensors and actuators[2] that are deployed to perform distributed sensing and actuation tasks linked by a wireless medium. The sensor nodes (small, cheap devices with limited computation) are deployed for the collection of data through a sensing mechanism, while actuators (resource rich, better processing capabilities and stronger transmission power) take decisions based on sensor input and then perform appropriate actions upon the environment, which allows a user to effectively sense and act from a distance. These networks can be the integral part of many military and civil applications such as disaster/crime prevention, real-time battle field screening, weather monitoring, environmental and health monitoring to smart spaces. For example, in case of fire, sensors relay the exact origin and intensity of the fire to water sprinkler actuators so that the fire can easily be extinguished before it becomes uncontrollable. Similarly, motion and light sensors in a room can detect the presence of people and then command the appropriate actuators to execute actions based on the pre-specified user preferences [2].

SANETs have the following unique characteristics:

- Real-time requirement: In SANETs, depending on the application there may be a need to rapidly respond to sensor input. Examples can be a fire application where actions should be initiated on the even area as soon as possible.
- Coordination: Unlike WSNs where the central entity (i.e., sink) performs the functions of data collection and coordination, in SANETs, new networking phenomena called sensor-actuator and actuator-actuator coordination may occur. In particular, sensor-actuator coordination provides the transmission of event features from sensors to actuators. After receiving event information, actuators may need to coordinate with each other (depend on the acting application) in order to make decisions on the most appropriate way to perform the actions.

Many protocols and algorithms have been proposed for WSNs in recent years [1]. However, since the above listed requirements impose stricter constraints, they might not be well-suited for the unique features and application requirements of SANETs. In [2], the authors present a comprehensive analysis on different types of coordination required for SANETs and several open research issues that should be investigated. Following the insight provided therein, we proposed the design of a self organizing framework for SANETs in [3]. To cope with the coordination challenges,

---

[1] The terms SANET and WSAN can be interchangeably used if SANETs are wireless.

[2] We refer to acting entities as actuators. They are also referred to as actors in related literature.

a self-organizing distributed coordination framework called ADP was proposed, and it was shown that the ADP provides optimal[3] routing paths to sensors. ADP is very simple to accommodate the limitations of small sensor nodes and is energy efficient. The functionality of ADP is independent of the routing protocol and is distributed to cope with large scale deployments.

In this paper, we present the performance analysis of ADP for both static and mobile SANETs. The ADP organizes a two tier heterogeneous network of sensors and actuators into clusters and each cluster is individually managed by an actuator. The initial routing paths provided by the ADP to the sensors depends on the outcome of a cost function, which is set to minimum hop routing for this study due to strict latency requirements for SANETs. If static routing is used, some paths will get more unitized as compared to the others. This may result in the early node deaths due to limited energy resource. Therefore, we modify the routing paths during the network lifetime based on the remaining energy on different routes, so that the lifetime of the network can be enhanced. The ADP is a novel self organizing framework with loose bonded three different coordination levels. We also present the three level coordination framework and application dependent classification of actuator-actuator coordination.

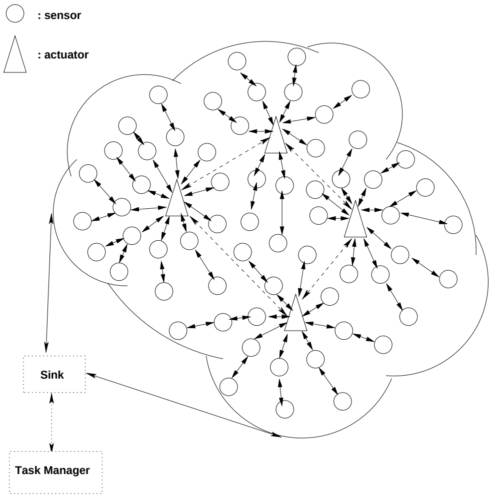

Fig. 1. SANET architecture

The remainder of the paper is structured as follows. In Section 2, We start by detailing the three level coordination framework. We then overview the ADP proposal and how it organizes the two-tiered network into clusters using distributed learning in Section 3. In Section 4, we present the network model. Section 5 details the power consumption model. Some implementation considerations are detailed in Section 6. The performance analysis of the ADP is presented in Section 7. In the end, we review some interesting research work in this direction in Section 8. Finally Section 9 summarizes the paper with a brief conclusion, and outline the future work.

2. A NOVEL THREE-LEVEL COORDINATION FRAMEWORK FOR SANETS

In SANETs, sensors acquire information such as light, temperature, noise, and humidity from the surroundings, while actuators take decisions based on the information received and perform relevant actions. An integrated support for data aggregation in such networks works flexibly well with all major aggregation proposals: diffusion algorithms [8], streaming queries [13], and event graphs [7]. The three approaches differ in the way they influence the energy utilization and delay constraints, so it is left as an application and requirements specific concern to be monitored by the actuators in the network. In general, the network may support a variety of task types. However, sensor-actuator networks are task-specific-unlike general purpose communication networks, the task types are known at the time a sensor-actuator network is deployed. We leverage this important observation in our design. We now focus on the three coordination levels of our self organizing framework.

- Sensor-sensor coordination level: In WSNs, in-network aggregation [8] and negotiation based routing schemes are shown to work in the absence of any architecture. Therefore, we consider the sensor-sensor coordination level as flat structured. The main problem with the flat architectures is its scalability to large deployments. In our considered architecture, this flat structure is locally applied for only sensor coordination in order to facilitate data aggregation functions. Further, the sensors need not to know about all the sensors that belong to the same cluster due to the existence of multi-hop paths and only neighbor knowledge is sufficient for effective coordination [3].
- Sensor-actuator coordination level: The coordination between sensors and actuators follows a hierarchical architecture, which has been shown to perform better [9], [11], [12], [13] in terms of defined QoS as compared to the flat architecture. To minimize the latency between sensing and acting, the main goal of this coordination is transmit the event information to the appropriate actuator in the shortest time. The excessive burden of relaying information to the actuators can cause the sensor nodes to die due to limited battery supply. Therefore, we also try to optimize the network lifetime by the introduction of an energy aware routing scheme at this coordination level.
- Actuator-actuator coordination level: The coordination

---

[3] Optimal here refers to the outcome of a cost-function in [3].

between the actuators follows a QoS architecture which can be divided into a number of categories based on application requirements [2]. As this particular coordination level is not constrained by limited resources, we can use AODV/OLSR like routing protocols for an efficient coordination among different actuators.

Optimizing all these coordination planes locally gives multidimensional improvements as a result of loose bonding between the three levels. This paradigm also removes the application dependencies from routing basics.

Application dependent classification of actuator-actuator coordination: When a sensor detect an event, it transmits the readings to an actuator node which can process the incoming data and initiate appropriate actions and is named as **Automated architecture**. The actuators can also route back the information to sinks which may issue action commands to actuators referred as **Semi-automated architecture**. We have modified the Automated architecture presented in [2] into two types which covers all the requirements for an effective actuator-actuator coordination.

- **Distributed Single-Actuator Automated Architecture:** A sensor transmits/forwards the readings to its optimal actuator. The actuator can process all incoming data and initiate appropriate actions, e.g., a high alert security application. The actuators can also route this information back to the sink for some remote processing. This approach can also be referred to as AF (Action First) approach. As shown in Fig. 2, a sensor detecting some event transmits this information to its closest actuator. The actuator react to this information with minimum latency according to the application requirements.

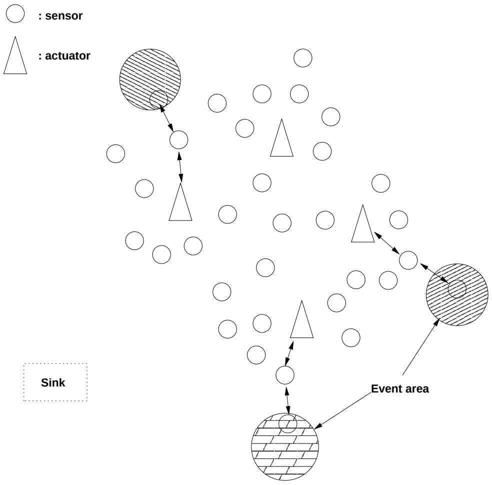

Fig. 2. Distributed Single-Actuator Automated Architecture

- **Distributed Multiple-Actuator Automated Architecture:** Upon receiving the event information, an actuator can it and route it to neighboring actuators in order to best decide the optimal actuation strategy, e.g., in case of fire, the actuators need to efficiently collaborate so that the fire can easily be extinguished before it becomes uncontrollable. In this fashion, an energy constrained sensor do not need to transmit its readings to multiple actuators as shown in Fig. 3. Instead, the first actuator to receive this event information will forward it to its neighboring actuators to come up with an optimal actuation plan. This approach can also be referred to as DF (Decision First) approach.

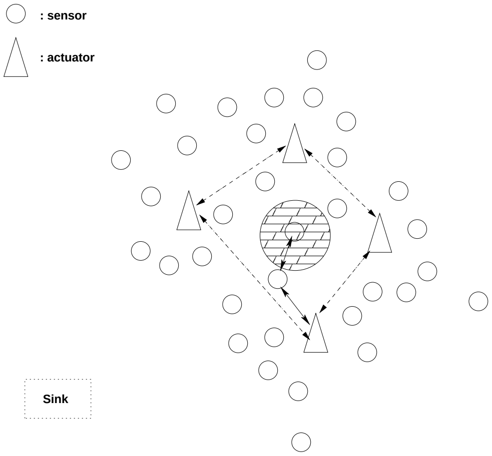

Fig. 3. Distributed Multiple-Actuator Automated Architecture

3. OVERVIEW OF THE ADP

ADP (Actuator Discovery Protocol, a controlled flooding mechanism [3]) is a discovery protocol that facilitates the first two coordination levels of our self organizing framework. A sensor node finds an optimal actuator using the proposed ADP, during the initial deployment phase. The sensors start the learning phase by transmitting a one hop broadcast actuator-search *request*, using their lowest transmit power. When a broadcast reaches an actuator, it is replied with the actuator identity. A random access scheme is used in the topology learning phase, because the sensors do not yet have a transmission schedule. The scheme is designed so that, at the end of this phase, almost all nodes are attached (based on minimum-hop routing) to an actuator and correctly determine their neighbors and interferers with high probability. We adopt a carrier sense multiple access (CSMA) mechanism similar to 802.11 [14]. The sensors listen for a random time before transmitting, and transmit if the channel is idle. A random delay is added before carrier sensing to further reduce collisions. However, because a collision will lead to incomplete cluster information at the actuators, the CSMA scheme itself cannot guarantee that an actuator will receive the full cluster information. Therefore, we proposed to include an implicit acknowledgment from the actuator, which occurs when a sensor transmits a packet to *join* a particular cluster. Based on this acknowledgment, a sensor selects a CDMA code to communicate with its actuator. The actuator explicitly schedules all the sensors, based on its knowledge of the cluster.

4. NETWORK MODEL

Consider a static wireless sensor-actuator network with n sensor nodes and m actuator[4] nodes as shown in Fig. 1. Given is an $(n+m) \times (n+m)$ neighborhood relation matrix R that indicates the node pairs for which direct communication is possible. We will assume that R is a symmetric matrix, i.e., if node i can transmit to node j, then j can also transmit to node i. For such node pairs, the $(i,j)^{th}$entry of the matrix R is unity, i.e., $R_{\{i,j\}} = 1$ if node i and j can communicate with each other; we will set $R_{\{i,j\}} = 0$ if nodes i and j can not communicate. For any node i, we define $N_i = \{j : R_{\{i,j\}} = 1\}$. Which is the set of neighboring nodes of node i. Similarly, a set of interference nodes (cannot be reached by one-hop) for node i (from where the transmissions can be heard at node i), and is defined as

$S_i = \{K \notin N_i \cup \{i\} : R_{k,j} = 1 \ forsome \ j \in N_i\}$

Note that S_i does not include any of the first-hop neighbors of node i.

The topology of the network is represented by a graph $G = (V,E)$, in which V is the set of nodes (both sensors and actuators). The edges $E \in V \times V$ are such that $(i,j) \in E$ if nodes i and j can transmit to each other.

5. POWER CONSUMPTION MODEL

For a sensor node, the energy consumption due to wireless communication (i.e receiving and transmitting) is considered the dominant source in power consumption. If power consumed to receive a single multi-hop packet (for design, we assume all packets to be of same length) is given by P_{rx} (in j/pkt), then the power consumed P_r^i (in j/s) by a sensor node i for receiving is

$$P_r^i = P_{rx} \sum_{j \in N_i} \alpha_{j,i} \tag{1}$$

where $\alpha_{j,i}$ is the rate (pkt/s) at which node j is transmitting packets toward node i.

If the power consumed to sense and sample a packet is P_{sense}(in j/pkt), then the power consumed P_s^i (in j/s) by a sensor node i in sampling packets is

$$P_s^i = P_{sense} \lambda_i \tag{2}$$

where λ_i is the rate (pkt/s) at which node i performs environmental sensing.

If the power consumed to send a packet is given by P_{tx}, then the power consumed P_t^i (in j/s) by a sensor node i in transmitting its data (both locally originated and forwarded packets) is

$$P_t^i = P_{tx} \sum_{j \in N_i} \alpha_{i,j} \tag{3}$$

When the packets arrive from S_i due to interference, the power loss P_{in}^i (in j/s) at node i is

$$P_{in}^i = P_{rx} \sum_{j \in S_i} \alpha_j \tag{4}$$

where α_j in (pkt/s) is the total rate at which node j is transmitting: $\alpha_j = \sum_{k \in N_j} \alpha_{j,k}$

If node i is neither serving its forwarding queue nor sampling a new packet, it is in idle state. If the power consumed in idle state is given by P_{idle}, then the power consumption P_{id}^i (in j/s) by a sensor node i is given by

$$P_{id}^i = P_{idle} \left(1 - \sum_{j \in N_i} \frac{\alpha_{i,j}}{C} - \sum_{j \in N_i} \frac{\alpha_{j,i}}{C} \right) \tag{5}$$

where C is the transmission capacity in $pkts/s$.

Network Lifetime $T_{lifetime}^{network}$ is defined as the time spanned by the network before first *node death* as a result of energy outage. The lifetime of a sensor node i having battery capacity E_i is given by (6).

$$T_{lifetime}^{network} = \min_i T_{life}^i \tag{7}$$

For the considered model, we use a TDMA-MAC protocol with minimized *awake periods* and avoid power loss due to *interference*. As the ADP provide each sensor with the shortest paths to reach an appropriate actuator, the lifetime T_{life}^i can be maximized by controlling the flow coming into a node using an adaptive routing protocol in the following fashion: Before transmitting data, a sensor node i computes the remaining power available P_j at all the uplink-neighbors, which is a simple ratio of the total energy consumed up to period p over the total battery energy:

$$P_j = p.\frac{Total\,EnergyConsumed}{Total\,BatteryEnergy} \tag{8}$$

In this way, a sensor node i picks an uplink neighbor j with the maximum value of P_j to prolong the network lifetime. As this computation consumes a significant amount of energy, a sensor node i will perform this computation

---

[4] Conceptually, we can assume that this actuator is also a sensor node, which has 0 sampling rate.

$$T_{life}^{i} = \frac{E_i}{\left(P_{rx}\left(\sum_{j\in N_i}\alpha_{j,i} + \sum_{k\in S_i}\alpha_k\right) + P_{tx}\sum_{j\in N_i}\alpha_{i,j} + P_{sense}\lambda_i + P_{idle}\left(1 - \sum_{j\in N_i}\frac{\alpha_{i,j}}{C} - \sum_{j\in N_i}\frac{\alpha_{j,i}}{C}\right)\right)} \quad (6)$$

periodically after some defined interval, which can be application dependent and decided *a priori*.

6. IMPLEMENTATION CONSIDERATIONS

In this section, we give the main assumptions that were considered during the implementation of the ADP protocol.

1) We consider a wireless ad hoc network that consists of a group of sensor and actuator nodes.
2) Sensors do the application dependent sensing and transmit their sensed data toward their optimal actuators.
3) Each sensor is equipped with a single omni-directional antenna except for the actuators which are equipped with two different antennas, one to coordinate with sensors and the other to communicate with neighboring actuators for a delay-efficient actuation process.
4) Both sensors and actuators are capable of adjusting their transmission power.
5) A link between any two nodes (sensors/actuators) i and j is bidirectional.
6) The actuators can reach all the sensors in their local cluster in only one-hop by using their maximum transmission capabilities.
7) The following metrics are most often used to compare sensor/actuator network coordination-protocols: energy consumption, delay and delivery ratio. Therefore, we evaluate only these metrics in this work.

7. PERFORMANCE EVALUATION OF ADP

We implemented the ADP in ns-2 (a standard Network-Simulator) [15] as an application layer protocol. In ns-2, there is no built-in support for simulating heterogeneous network topologies. In order to simulate a two-tiered network of distributed sensors and actuators, we post process the tcl-scripts containing the topology information during the learning phase, make a few nodes (percentage decided a priori) as actuators, and modify their communication capabilities. The sensor data is first gathered at these actuators nodes in their respective clusters and then forwarded to sink(s) using the actuator-actuator coordination. For sensor-sensor coordination, the sensors only require one-hop neighbor identity through which it can reach the actuator with lower cost as compared to its own. For sensor-actuator coordination, we simulated topologies of various sizes (50-400 sensors). The considered packet size is 50 bytes and the transmission rate is 50kpbs. Shortest path routing is used in the simulations. The average depth of the resulting routing trees is 4.4, 5.2, and 7 for 20, 30, and 60 sensors per cluster, respectively; correspondingly the average number of neighbors is 4.6, 5.0, and 5.5. The transmission range is 100 m and the initial energy in the sensors is 1000 j. Other simulation parameters are listed in Table I.

TABLE I. Simulation Parameters. The simulation ares is set such that there are atleast two sensors in each others transmission range.

Sensors	Area(m^2)	Actuators/Sinks
50	350 * 350	2
100	500 * 500	2
150	600 * 600	3
200	700 * 700	4
250	790 * 790	5
300	860 * 860	6
350	920 * 920	7
400	970 * 970	8

Directed Diffusion [8] and anycast [7] is chosen as the routing protocol for comparison. IEEE 802.11 is chosen as the medium access control (MAC) protocol. Simulation time is 1001 s which is sufficient to characterize protocol trends. During the initial 50 s, the nodes gather information about one-hop neighbors and their attached actuators as explained in [3]. Afterwards the sensors generate data every 3 seconds. Actuators[5] move at speeds of 1, 3 5, 8, 10, 15 and 20 m/s, respectively. Whereas sensors move only at 1 and 2 m/s. We have simulated the distributed single-actuator automated architecture, so there is no interest-propagation implemented for the moment (which is a requirement for distributed multiple-actuator automated architecture). There is no actuation mechanism implemented in the current ns-2 simulation. The estimated delay is the end-to-end delay seen by a transmission.

Figure 4 shows the end-to-end latency as a function of network size. The delay increases with the increase in the network size, but the increase is significantly less for ADP. This gradual increase is the result of smaller mean-path length for ADP as the cost-function is set to min-hop routing and forwarding queues at the sensors are not saturated at the given load.

Figure 5 and 6 show the mean energy consumption as a function of time. ADP energy savings are more significant due to the existence of multiple defined routing paths toward

[5] Actuators/Sinks have similar semantics for simulation purposes.

optimal actuators, where depending on the remaining energy of the forwarding sensors, a source can choose between several available paths to efficiently route its data.

In Figure 7, the mean path length is shown as a function of network size. Again the mean path length (which is related to the end-to-end latency) increases with the network size. However, the increase is more gradual with the ADP as compared to anycast and directed diffusion. Using ADP, sensors always transmit their data to the nearest actuator (because we set the cost-function to min-hop routing for actuator discovery during initial deployment).

Figure 8 shows the delivery ratio as function of speed of actuator node for a topology of 100 sensors. This mobility scenario can be seen as static sensors and mobile actuators where ADP achieves 99% delivery rate at actuator speed-5 m/s, and 98.5% delivery rate at the actuator speed-20 m/s. This comes from the fact that sensors require one-hop local message exchange in mobility scenarios to update their routing tables [3].

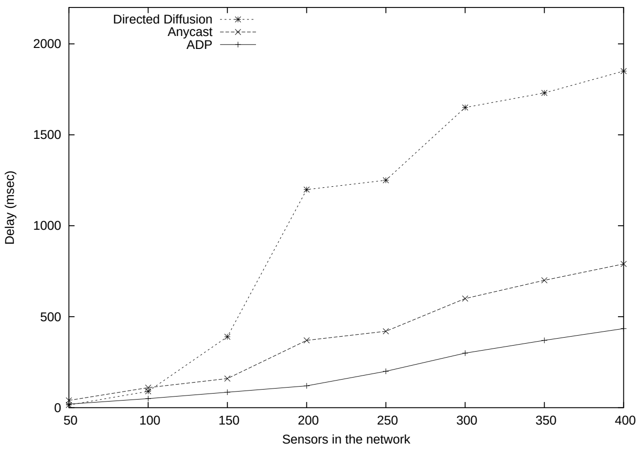

Fig. 4. Mean end-to-end transmission delays

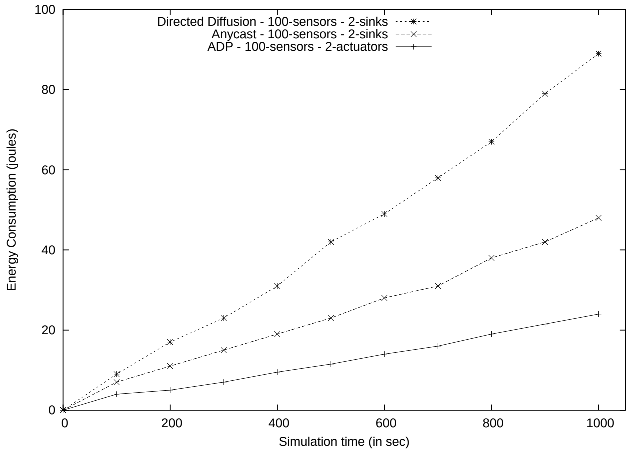

Fig. 5. Mean energy consumption as a function of time for a network of 100 sensors

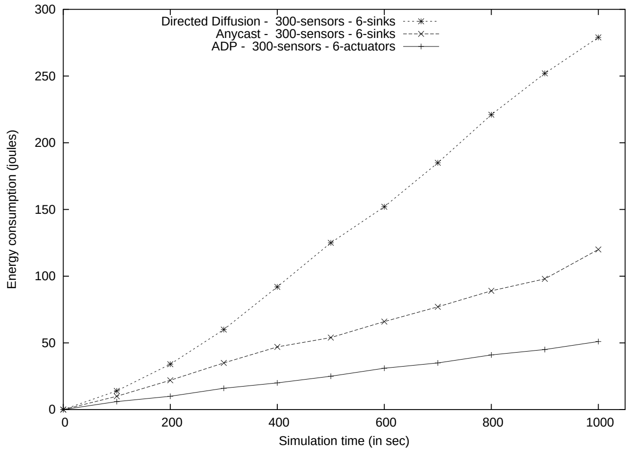

Fig. 6. Mean energy consumption as a function of time for a network of 300 sensors

Figure 9 shows the increased energy consumption to update the broken paths as a result of mobility, where the update requires only one-hop messages among neighboring sensors.

In a target tracking application, a highly mobile target moving at the speed of 120 km/h travels around 433 meters within 13 s. To demonstrate the delay experienced in course of event mobility, we enabled upto thirty sources to send a packet toward their attached actuators with a difference of 30 *msec* between each transmission. The observed delay should be bounded such that the actuators can perform any degree of actuation on the intruder. Figure 10 shows the delay experienced by the transmissions in the case of event mobility. We have obtained the delay as low as 10 *msec*, which comes from energy balanced min-hop routing in the design of ADP. The increase in delay with the increase in number of concurrent sources (transmitting sensors) is due to TDMA like MAC with fixed scheduling slots.

8. RELATED WORK

In [8], the authors proposed an efficient routing protocol for wireless sensor networks (WSNs) with global objective set to maximize network lifetime. The constraints are set to minimize the energy consumption for efficient data aggregation. The protocol works by building gradients along an interest propagation. In short, interest propagation sets up state in the network (or parts thereof) to facilitate "pulling down" data toward the sink. The results provided therein have shown significant improvement over traditional routing protocols both in terms of communication and computational load. Whereas in [7], the authors use the same approach as [8] for sensor-actuator networks using anycast routing.

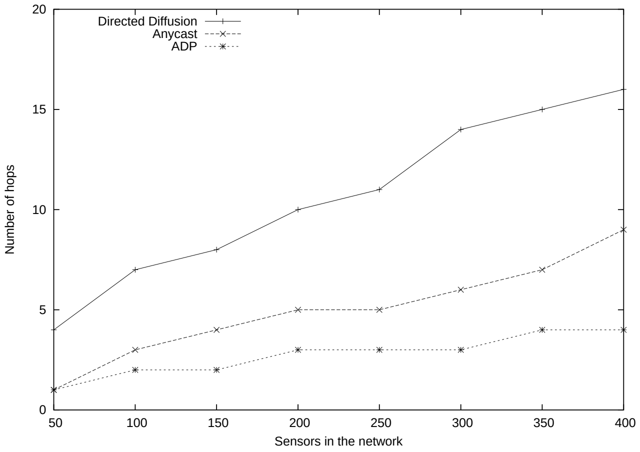

Fig. 7. Mean number of transmissions per end-to-end path (mean path length)

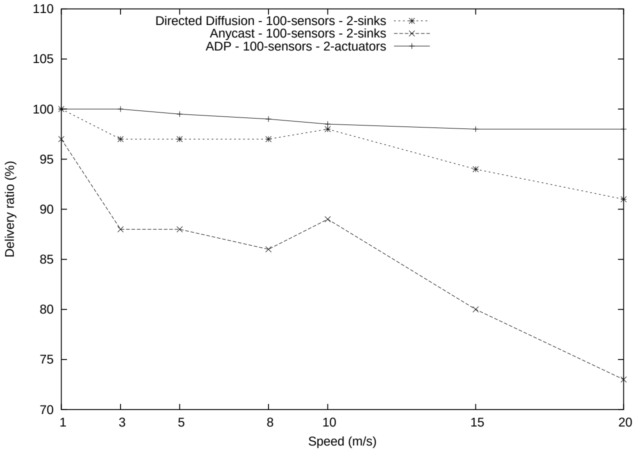

Fig. 8. Date Delivery rate: where actuators/sinks move at variable speed

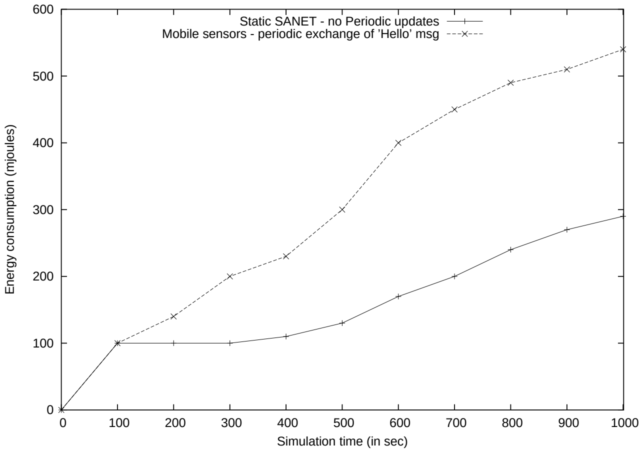

Fig. 9. Energy consumption for control overhead in ADP

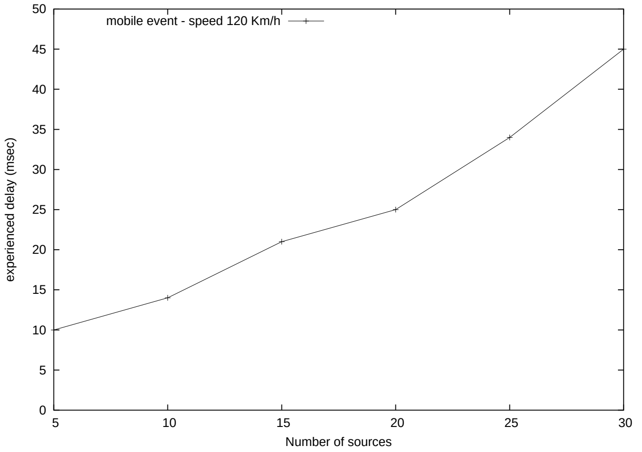

Fig. 10. Delay experienced by mobile event

A reverse tree-based anycast routing is proposed, which constructs a tree routed at the event source, where sensors can join and leave dynamically. The introduction of actuators in the existing wireless sensor networks has opened up a new dimension of "a hard delay constraint" while still looking for near-optimal network lifetime solutions [2]. For example, Targeting an intruder holding a sniper in a surveillance field can be an interesting case to consider. The actuation process has to localize the position of the intruder and actuate the destruction process. The important constraint in this case is the latency of the received data because the sensor data can be no more valid at the time of actuation in case of increased latency.

A well designed application-specific coordination protocol is proposed [6], where cluster formation is triggered by an event so that clusters are created on-the-fly to optimally react to the event itself and provide the reliability with minimum energy expenditure. In order to provide effective sensing and acting, an efficient and distributed coordination mechanism is required for delay-energy aware dissemination of information, and to perform right and timely actions. Therefore, we proposed to establish these clusters once during the initial network deployment and the routing protocol can disseminate the sensed information to the actuators through maximum remaining energy paths. After receiving the event information, actuators may need to coordinate with each other in order to make decisions on the most appropriate way to perform the required action. Depending on the application, there can be multiple actuators interested in some information. Therefore, sensors need to transmit this data toward multiple actuators, which results in excess sensor-energy drain due to multiple transmissions of redundant information [5]. Moreover, the collected and transmitted sensor data must be valid at the time of acting. For example, if sensors detect a malicious person in an area and transmit this information to its optimal actuator; and the act of disposing a tranquilizing gas must find that person in the very same area. Therefore, the issue of real-time communication is very important in SANETs.

Most of the current research on sensor systems is mainly focused toward optimizing the network lifetime (e.g., [4]) and the energy consumption of the sensors bypassing the delay-sensitivity of sensor data for real time applications. In [10], the authors presented a detailed overview of the routing techniques proposed for WSNs. The routing techniques are classified into three categories based on the underlying network structure: flat, hierarchical, and location-based routing. The hierarchical routing schemes have shown a promising improvement for prolonging network lifetime [9]. An enhancement in basic LEACH is proposed in [11], where the network lifetime has been extended by the introduction of closest neighbor communication. In [12], [13], the network lifetime was prolonged on the basis of threshold-sensitive routing schemes. All of these protocols share a common problem: routing semantics binded to application requirements.

Note that in this work, we have analyzed the performance of a sensor-actuator network architecture through ns-2 simulations which is robust and entirely distributed with three distinct coordination phases. We have also shown that the application dependent coordination semantics should be handled independent of routing dynamics.

9. CONCLUSION AND FUTURE WORK

Routing in sensor-actuator networks is a new research area, with constrained, but rapidly growing set of research results. In this work, we evaluate performance of our proposed low-energy adaptive and distributed coordination protocol for both static and mobile networks. For sensor-actuator coordination, the proposed ADP can acquire a promised QoS for time stringent traffic at the cost of optimal energy consumption for both static and mobile networks. The issues related to mobility are well handled by the energy-efficient (only among one-hop neighbors) periodic exchange of ’Hello’ messages. The distributed learning proposed in [3] also minimizes the routing table size at each sensor node.

We are working on the implementation of distributed multiple-actuator automated architecture, where the actuators will diffuse their interests in the network at deployment time and an actuator will coordinate with its neighbors upon receiving an event information to decide an optimal actuation strategy. We are also working on the integration of a TDMA like MAC protocol with embedded synchronous wakeup protocol to improve network lifetime, transmission capacity and delay bounds. The TDMA MAC and multiple-actuator architecture will be implemented in TinyOS and EmTOS using the TOSSIM [16] and EmSim [17] simulators for the support of heterogenity.

ACKNOWLEDGEMENTS

Institut Eurécom’s research is partially supported by its industrial members: BMW Group Research & Technology - BMW Group Company, Bouygues Télécom, Cisco Systems, France Télécom, Hitachi Europe, SFR, Sharp, STMicroelectronics, Swisscom, Thales.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci. 2000. “Wireless sensor networks: A survey.” Computer Networks, 38, No. 4, 393-422.
[2] I. F. Akyildiz and I. H. Kasimoglu. 2004. “Sensor and actor networks: research challenges.” Ad Hoc networks (Elsevier), 2 No. 4, October: 351-367.
[3] M. Farukh Munir and F. Filali. 2006. “A novel self organizing framework for SANETs.” In Proceedings of the 12th EW Conference (Athens, Greece, Apr. 2-6).
[4] V. Paruchuri and A. Durresi. 2005. “Delay-energy aware routing protocol for sensor and actor networks.” In Proceedings of the Icpads Conference (Fukuoka, Japan, Jul. 20-22). 292-198.
[5] E. Cayirci, T. Coplu and O. Emirogl. 2005. “Power aware many to many routing in wireless sensor and actuator networks”. In Proceedings of the Ewsn Conference (Berlin, Germany, Jan. 31-2 Feb.). 236-245.
[6] T. Melodia, D. Popili, V. C. Gungor and I. F. Akyildiz. 2005. “A distributed coordination framework for wireless sensor-actuator networks.” In Proceedings of the MobiHoc Conference (Urbana-champaign, IL, USA, May. 25-28). .
[7] W. Hu, N. Bulusu, and S. Jha. 2005. “A communication paradigm for hybrid sensor-actuator networks.” International Journal of Wireless Information Networks. 12 No. 1. January:47-59.
[8] C. Intanagonwiwat, R. Govindan, and D. Estrin. 2000. “Directed diffusion: a scalable and robust communication paradigm for sensor networks.” In Proceedings of the ACM Mobicom Conference (Boston. MA. USA Aug. 6-11). 56-67.
[9] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. 2000. “Energy-efficient communication protocol for wireless microsensor networks.” In Proceedings of the Hicss Conference (Maui. Hawaii. Jan. 4-7). .
[10] J. N. Al-Karaki and A. E. Kamal. 2004. “Routing techniques in wireless sensor networks: a survey.” Article wireless communications. 11. No. 6. December: 6-28.
[11] S. Lindsey and C. Raghavendra. 2002. “PEGASIS: power-efficient gathering in sensor information systems.” In Proceedings of the Acp Conference (Mar. 9-16). 3(1125-1130).
[12] A. Manjeshwar and D. P. Agarwal. 2001. “TEEN: a routing protocol for enhanced efficiency in wireless sensor networks.” In Proceedings of the Wnmc Conference (Mesa. Ar. USA. Apr. 17-19). 2009-1025.
[13] A. Manjeshwar and D. P. Agarwal. 2002. “APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks.” In Proceedings of the Ipdps Conference (Fort Lauderdale. Fl. USA. Apr. 15-19). 195-202.
[14] LAN-MAN Standards committee of the IEEE Computer Society. 1997. "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification," IEEE Std 802.11-1997.
[15] The Network Simulator - ns (version 2.27). http://www.isi.edu/nsnam/ns/.)
[16] http://www.tinyos.net/
[17] http://cvs.cens.ucla.edu/emstar/