Institut Eurécom
Department of Mobile Communications
2229, route des Crêtes
B.P. 193
06904 Sophia-Antipolis
FRANCE

Research Report RR-06-158

An Energy-Aware Actuator Discovery Protocol for SANETs

March 30th, 2006

Muhammad Farukh Munir and Fethi Filali

Tel : (+33) 4 93 00 82 03
Fax : (+33) 4 93 00 82 00
Email : {Muhammad-Farukh.Munir, Fethi.Filali}@eurecom.fr

---

[1] Institut Eurécom's research is partially supported by its industrial members: Bouygues Télécom, Fondation d'entreprise Groupe Cegetel, Fondation Hasler, France Télécom, Hitachi, Sharp, ST Microelectronics, Swisscom, Texas Instruments, Thales.

An Energy-Aware Actuator Discovery Protocol for SANETs

Muhammad Farukh Munir and Fethi Filali

Abstract

SANETs (Sensor-Actuator networks) are appealing to researchers due to their wide spread of application potential, ranging from densely deployed habitat-monitoring setup to the real-time security applications. This paper presents the design, implementation and performance evaluation of a novel energy-aware coordination framework for SANETs. We assume two wireless interfaces at the actuator nodes, one to communicate with sensor nodes, and the other, for the neighboring actuator nodes for fast and effective actuation process. We design a new protocol called ADP (Actuator Discovery Protocol) which provides an optimal attachment for a sensor node to the actuator with minimum cost. For the sensor-actuator coordination, the proposed ADP can guarantee ordering, synchronization and eliminates the redundancy of actions. A distinct feature of ADP is that its performance is robust to operating topologies, energy consumption and delay constraints. Simulations using Network Simulation (ns-2) show how global network objectives, such as compliance with real-time constraints, minimum energy consumption, mean number of transmissions, and load-balancing among local-clusters, can be reached in our proposed framework with simple interactions between sensors and actuators that are suitable for any-scale networks of energy-constrained sensors.

Keywords: SANETs (Sensor and Actuator Networks), self-organization, coordination, energy-aware protocols.

Contents

List of Figures

1 Introduction

The advent in technology led to the emergence of SANETs (Sensor and Actuator Networks) giving a distributed control to the management, communication and coordination aspects of the network functioning formerly referred to as WSNs (Wireless Sensor Networks). A SANET consists of a group of sensors and actuators, that are deployed to perform distributed sensing and actuation tasks, linked up 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 and then perform appropriate actions upon the environment.

Current applications for WSANs are mostly research prototypes or are tailored toward a specific purpose. There is no typical WSAN structure and architecture, and the basic goals of a WSAN rely mostly on the considered application. Most of the sensor network research tends to focus on specific hardware with efficient, ingenious communication protocol algorithms and system control architectures that address the specific resource constraints, which did not lead to the emergence of a generic architecture for the sensor networks [1, 2].

Despite of the sufficient work done in SANETs [3, 4, 5, 6], an effective coordination mechanism for efficient sensing, dissemination of information, and to perform right and timely actions (actuated by the actuators) is required. These networks can be the integral part of systems such as: disaster/crime prevention, real-time military applications, environmental and health monitoring to smart spaces [7].

This paper investigates a new self-organizing framework for the SANETs. In this regard, we design a new self-organizing protocol called ADP (Actuator Discovery Protocol) which provides optimal-attachment to any sensor node in the network with an optimal and most relevant actuator node. We have determined a generalized cost function to derive the optimal paths toward the actuators and hence an optimal actuator (detailed in Section 3.2). Therefore, a sensor node anywhere in the network finds the optimal actuator using the proposed ADP during the initial deployment phase. Afterwords the data is relayed to the attached-actuator through a multi-hop path unless the network undergoes a remarkable topology change, which covers the basics of sensor-actuator coordination. 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. For the actuator-actuator coordination, we employ the use of the assumed separate wireless interface to communicate with the neighboring actuator so that they can perform long-range data communication without any involvement of sensor nodes to effectively coordinate and to perform the actuation process. The examples of actuation cover a wide range of possibilities and are typically application dependent. For example, Targeting an enemy holding a snipper in a battle field can be an interesting case to consider. The actuation process has to localize the position of the enemy and actuate the destruction process. But the important constraint in this case is the latency, the sensor data can be no more valid at the time of actuation in case of increased latency. Which makes the assumption of a separate wireless interface for actuator-actuator coordination quite relevant. Moreover, the sensor nodes need not to find the actuator all the times when it has some sensed data to transmit for reasons detailed in section 3. Likely, we can infer the working of ADP for a wide range of possibly existing scenarios.

The proposed solution consists of three phases; namely Learning, Coordination, and Failure and Recovery phases.

The Learning-phase starts during the initial deployment, as depicted in Figure 1, when the sensor nodes try to locate the nearest actuator using a broadcast strategy through all the neighbors that are in their sensing radius. Then it chooses the optimal-path and keep it for future use. The actuator node also keeps the copy of the shortest path to any sensor node in its cluster. For more detailed explanation on coordination, and failure and recovery phases, see [19].

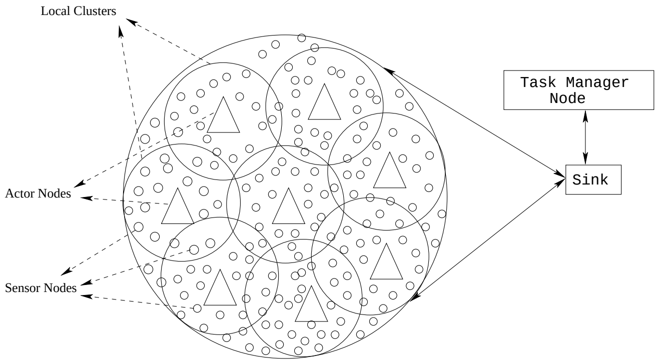


Figure 1: Architecture of SANETs

The remainder of the paper is structured as follows. In Section 2, we review the related work. In Section 3, the self-organization framework and the dynamics of the proposal are explained in detail. A performance analysis with prior work is presented in Section 4. Section 5 details the simulation results and evaluation of the proposed protocol. Section 6 presents a qualitative analysis of the proposal against different deployment scenarios to testify its efficiency. Section 7 terminates the paper with a brief conclusion.

2 Related Work

Recently a few research communities tried to explore the potential benefits of SANETs over WSNs, proposed distributed approaches to minimize the energy consumption, and explored various potential research fields [3, 4, 5, 7]. A new any-cast mechanism is proposed [6] for data dissemination to reduce the energy burden on the neighboring nodes. The authors have shown that the goal is to reach the nearest actuator through an any-cast scheme to conserve energy. The construction of an any-cast tree routed at the event source proves to work better compared to diffusion and other prior work but it is not the optimal solution to conserve energy and no bounds can be obtained on the max-latency in case of large networks. Comparing our work with another event driven clustering mechanism [5], shows that the coordination framework proposed by the authors works well in the environment where events are detected rarely. If the frequency of these events tend to grow, both the energy consumption and latency for the sensor increases sharply and the phenomenon can be easily observed by taking large networks into consideration. Researchers have so far mainly focused on picking up a particular deployment of sensor networks and addressed relevant problems (routing in terms of both energy-efficiency and real-time constraints) but none tried to work out a typical WSAN structure and architecture. A number of routing protocols have been proposed to reduce the energy consumptions of the sensors nodes [1, 2]. In [8], a minimum cost heterogeneous WSN is proposed to reduce the amount of residual energy in the network.

Most of the work done yet have taken into consideration the GPS (highly inefficient for energy considerations) for localizing the positions of the sensor nodes to accordingly, but in [9, 10] the authors have utilized an MDS based localization mechanism to obtain the approximate positions of the sensor nodes. A wiring mechanism is proposed for effective energy utilization through the introduction of wires at relevant points in the network. WSNs are envisioned to be used for a diverse range of distributed applications, ranging from densely deployed habitat monitoring to stern real-time military and/or security applications [11]. Correlations (spatial and temporal) play an important part toward the global energy utilization of the WSNs [12, 13, 14]. An MPC (minimum power configuration) approach is used to integrate the topology control, power-aware routing and sleep management as a joint optimization problem which is shown to be inherently dependent on the data rates of the sources [15]. In [16], they have established a synchronized sleep schedule based on asymptotically optimal clustering in poly-logarithmic time without relying on the MAC support. Adding mobile sensors to improve the multi-path connectivity (k-connectivity) is desirable to provide a certain defined degree of fault tolerance [17]. The results shown that the problem can be further improved to find a polynomial-time approximation. A mobility pattern is worked out to find the intruders (military/security applications) and the mobility is believed to be exploited for the lack of sensors and improve coverage [18].

The situation of sensor and actuator networks is exactly analogical to the situation of Internet approximately 30 years back. Note that in this work, we study the generic architecture and coordination-mechanism which is robust and entirely distributed with a little varying-degree of centralized control for the actuator on their local clusters. Instead of trying to achieve improvements only for a particular configuration, we have taken into consideration a more general approach to-wards optimizing the energy consumption and a promising delay to cope with the wider diversity in the field of SANETs. We try to show by examples that the earlier work done is mainly focused on particular application and not for the generic SANET architecture.

3 The Novel Actuator Discovery Protocol

As we have already explained in the earlier section that there is no sensor/actuator network structure and architecture, and the basic goals of the previously done work relied mostly on the considered application. In this work, we propose a coordination protocol which is tailored toward a standard behavior for most deployment scenarios, aiming to satiate the time-stringent requirements and effective energy utilization in a purely distributed fashion. In the following sub-section we detail the working of ADP.

3.1 The Governing Dynamics

The coordination protocol starts governing the dynamics of the communication when the sensors during the initial deployment stage locate the nearest actuator using a one-hop broadcast. The deployment of the sensor and actuator nodes can be randomly chosen or it can follow any pre-defined distributions. The finding of the “optimal-actuator attachment” for each sensor node is done through a novel protocol called ADP.

When a sensor node is turned on, it should first determine to which actuator node it has to be associated. Now, if we look at the execution of Algorithm 1, each node starts by broadcasting “AttachRequest” to the neighbors in its sensing range as shown in Figure 2. If this request is received by an actuator node, it replies with its own identity to the node from where the search prove arrives. A neighboring sensor upon receiving an AttachRequest message checks that it has sent an AttachRequest in the period

T_n (pre-configured timeout), if it has already sent an AttachRequest to its neighbors either to discover its own actuator or a forwarded message received from other neighbors, it has to wait for a reply until the defined timeout. Otherwise it does the same unless the probe reaches the nearest actuator. The reply message named "AttachReply" from the actuator follows the reverse path of the "AttachRequest" and terminates at its origin defining a discrete path to the sensor node. If an actuator node receives multiple AttachRequests from the same sensor node, it is bound to reply with the supposition that all these AttachRequests follow a different routing path to the actuator and the path decision is left to the sensor node. The sensor nodes can receive multiple replies from the neighbors along with possibly different actuator replies, it determines its optimal actuator and the path will be used to reach this actuator.

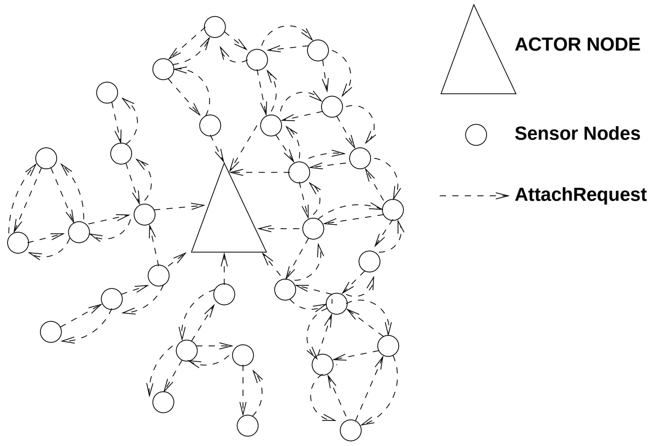


Figure 2: AttachRequest by sensors at the start of ADP

If a sensor receives multiple paths (containing same or different actuator-ID) then it chooses the optimal one (hop-count), and certainly if it receives more than one path containing the same hop-count, it chooses the earliest reply. Then it sends a "JoinRequest" to-wards this actuator. The actuator nodes upon receiving the "JoinRequest" sends the respective sensor node an "JoinAck" packet to confirm the joining in its local cluster and establishes itself an SOT (Self-Organized Tree), see [19].

As depicted by Figure 3, now each sensor node has a specified path to route its sensed data to its actuator nodes by simply forwarding it to only one of its one-hop neighbor (immediate next node in the path to the actuator), and the actuator also keeps the defined path to the node (building its tree structure for the localized cluster).

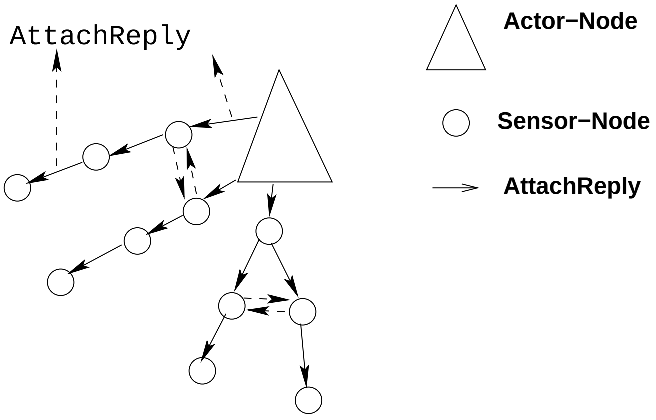


Figure 3: Actuator-replies (AttachReply) for corresponding AttachRequest messages

3.2 Protocol Control Semantics

In Algorithm 1, we have induced a cost function (control procedure) to obtain a promised QoS in terms of delay and energy consumption. For this work, we have assigned the hop-count to this function to restrict the probe from reaching the other end of the deployment by limiting the probe packet not to cross a certain hop-count (referred as 'C'). This hop-count can be treated as a function of sensor-actuator node ratio in the network to limit unnecessary broadcast and also keeping the chances of actuator-discovery well alive (leaving the issue as implementation concern and not the design one, deployment specific). This probe control is not practically visible because of the probability for this situation to arrive being extremely low (see Section 3.1). In general, limiting the broadcast packets to a certain number of hops results in energy conservation. We don't take into account the distance between any node and its associated neighbors with the reason being that the energy required to transmit to a node in its sensing radius is a constant (no power control assumed for transmissions).

Algorithm 1 ADP

```
Pseudo-code executed by all the sensor nodes $N_i$ during initial deployment-phase.
Initially:
 MinCost = $\infty$
 attached-actuator = $\infty$
 C = constant (the trade-off is explained in Section 3.2).
 $A_i$ = Identity of the Actuator.
 For any sensor node $N_i$
 do $ActorDiscovery$() {
    if cost ($N_i, A_i$) = $\infty$ then
     for each neighbor $M_j$ of $N_i$ do
       Send $AttachRequest$($cost, M_j, C$)
       Receive $AttachReply_i(cost, M_j, A_i)$
       #Determine optimal Actuator, and the next-hop among neighbors to reach it.
         for each $AttachReply$ do
           if path($cost, M_j$) < MinCost then
             $N_i$ MinCost = path($cost, M_j$)
             attached-actuator = $A_i$
             next_hop_to_actuator = $M_j$
           end-if
         end-for
     end-for
    end-if
}
After deciding the actuator, each node sends a "JoinRequest" to the its actuator.
    send JoinRequest($A_i$)
The actuator sends an "JoinAck" back to the sensor node confirming cluster joining.
    send JoinAck($M_j$ , $N_i$)
The procedure AttachRequest is implemented recursively as follows.
  $AttachRequest$($cost, M_j, C$){
    if ($cost$ != $\infty$)
      return (UpdateCost($cost$), $M_j$, $A_i$ )
    else if (C != 0) then
      for all neighbors $M_j$of $N_i$
        do $AttachRequest$($cost, M_j, C-1$)
      end-for
    end-if
}
Actuator Reply to the "AttachRequest" messages from the one-hop away nodes contains the
following.
  $AttachRequest(cost, M_j, C) \leftarrow$ $ActorReply$(cost = 1, $A$)
UpdateCost() is the part of the control semantics, and for this specific case it is chosen to
be hop-count.
UpdateCost(cost) {
  return cost + 1
}
```

The background for this consideration takes us to the lack of a GPS for position localizations in our study, which eats up a huge amount of sensor nodes' energy. In a similar fashion all the nodes reserve an optimal path to their optimal actuator as shown in Figure 4, forming a local cluster, thus giving us the initial deployment in the form of well-distributed small clusters. This also gives us a good degree of distributed control over the sensor nodes in the local clusters. ADP produces loop-free paths to the actuator nodes: *The next-hop selected by any sensor node with ADP has a defined optimal path to the actuator node (see Algorithm: 1).*

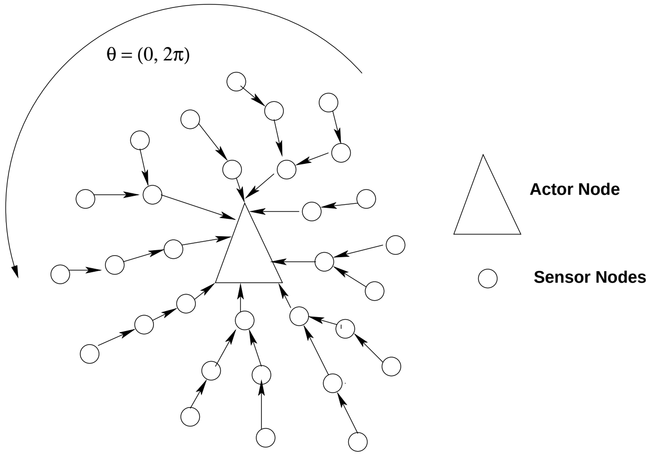
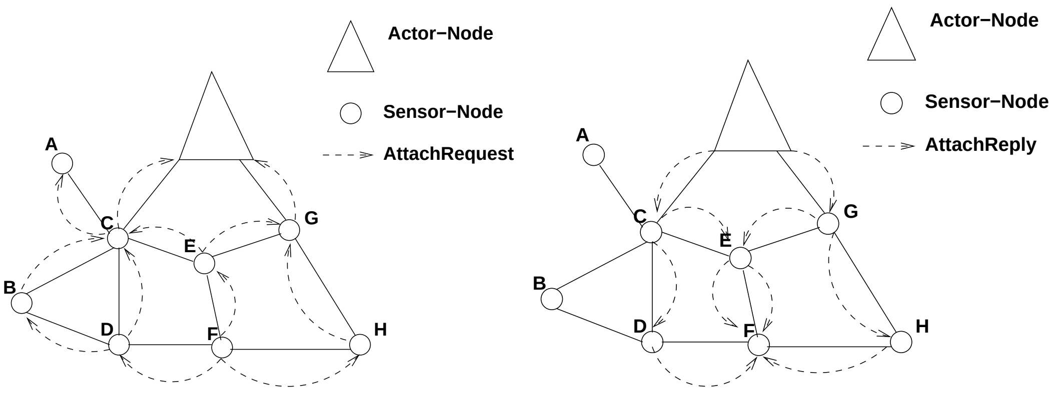


Figure 4: The Local-Cluster formulated at the termination of ADP

The deployed sensor nodes start sensing the distributed environment, and transmit their data through the defined path to the attached actuator. For the sensor-actuator coordination the actuator-attachment and the paths obtained to route data to the actuator provide effective energy optimization for the sensor nodes. For the actuator-actuator coordination we have utilized another wireless transmission interface to keep information about the neighboring actuator and operating at a higher degree than sensor/actuator coordination. The actuator-actuator coordination has to be fast enough for the timely-actuation process. In this fashion, the selection of single or multiple actuators for the actuation, in-sequence delivery of different events detected in a region along with a varying degree of challenges [3] are handled quite effectively. The actuators route their data to the sinks using a MANET routing protocol such as AODV (Ad Hoc On-Demand Distance Vector routing) and OLSR (Optimized Link State Routing) to confront with the issues related to the validity of sensor data at the time of actuation. The motivation behind choosing these protocols as examples of routing protocol for the actuator/actuator coordination is that the network of actuators could be seen as a Mobile Ad-Hoc Network (MANET) which in general has a low mobility. This whole strategy helps to actuate the required actions in an energy efficient and timely-coordinated manner with a guaranteed QoS (application specific approach). There can only be two deployment configurations for the SANETs: static and dynamic deployment. In this study we present our detailed analysis for the static case. In this deployment, both the sensor and actuator nodes are static. The gain with the static distribution is the maximum one due to efficient dissemination of routing data to the acquired actuators also depending on the environment friendliness of the deployment and no accidental node failures. The gain in the efficiency can be intuitively seen by the trade-off between the time required in network-learning and then adapting it. Monitoring the penetration of the discovery-probe gives a good balance for energy-efficiency and a predefined guarantee for the actuator-discovery.

3.3 An Illustration Example

In Figure 5, we consider a network with one actuator and 8 sensors. Lets consider as an example a sensor "F" to see how the sensor discovers its associated actuator. As shown in Figure 5(a), the sensor nodes starts ADP by broadcasting an "AttachRequest". The "AttachRequest" from sensor "F" is received by the sensors: 'D', 'E', and 'H'. Lets assume that they have not yet sent the "AttachRequest" message. All these receiving nodes then forward this "AttachRequest" to their neighbors. If the neighbors turns out to be the original sender of the "AttachRequest" message the will simply drop this packet at reception. The sensor 'D' sends the "AttachRequest to sensors 'B' and 'C', sensor 'E' sends it to 'C' and 'G', and sensor 'H' sends it to 'G'. From sensors 'C' and 'G', the "AttachRequest" reaches the actuator, or if they have already sent the "AttachRequest", the "AttachReply" would have been received by the arrival of this probe, or finally the sensors are still waiting for the "AttachReply" untill the predefined timeout expires. From the actuator node, the "AttachReply" message follows the path back to the origin of the packet. It is evident from Figure 5(b) that the sensor 'F' receives 4 different paths in this approach, and it is the choice of the sensor node to decide the routing path for its future correspondence with the actuator nodes (based on the cost-function).



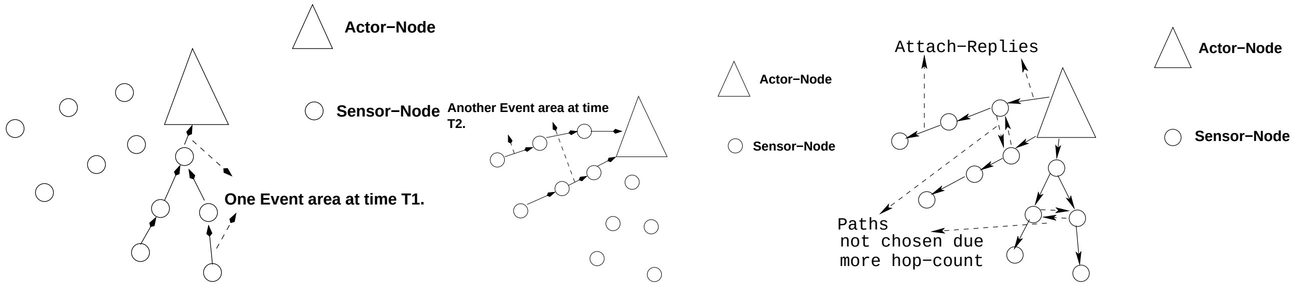
(a) Energy-efficient Controlled Boradcast of AttachRequest messages

(b) AttachReplies following the reverse path to the Originators of AttachRequest

Figure 5: An example to illustrate the working of ADP

4 Qualitative Analysis of ADP

In order to qualitatively evaluate the performance of our approach, we have selected the most relevant work done in WSANs [5]. The authors explained that, sensors detecting an event coordinate with each other to optimally associate each sensor with an actuator. In this fashion, only the event area is clustered and each cluster consists of those sensor nodes that send their data to the same actuator. Hence, the event information is collected at the optimal actuator nodes while existing energy resources are better utilized. Clusters are formed only when necessary, based on the event and the position of the actuators.



(a) Event detected by a group of Sensors at T1

(b) Event detected by another group of Sensors at T2

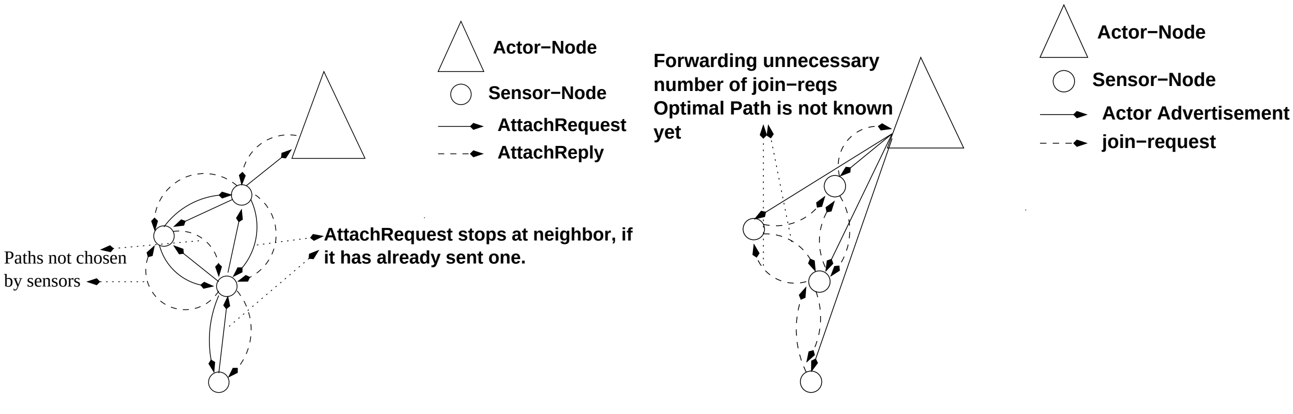
(c) Sensor Nodes in the Local Cluster of an Actuator

Figure 6: ADP Vs. event driven clustering

In the Figure 6, the sub-figures (a) and (b) are the two different event driven clusters for the same network at different times T_1 & T_2 respectively. In both of these events the closest found actuator for two different node-sets (sensors) ended up to be the same. The authors assumed to have eliminated the communication overhead to maintain clusters before the event occurs, which is desirable especially in application scenarios when events are rare. Now lets suppose that, these sensor nodes could be among an actuator's local cluster obtained by ADP as shown in Figure 6(c). With ADP, after the learning-phase is over, all the sensor nodes have defined routes to reach the optimal actuator. Therefore, no further energy is consumed in finding the next hop, to forward the sensed data, to-wards the optimal actuator as with event driven clustering. Latency is also reduced to practically cope with any desired level of actuation process. ADP seems to perform better with a wider range of network topologies. As mentioned before, the prior work done in the field of SANETs was tailored toward a specific application scenario and not toward a generic approach. For event driven clustering, if the number of sensed events starts to increase, the energy consumption of the sensor nodes increases exponentially in case of large networks due to the existence of more hops between sensors and their associated actuators.

In order to further validate the effectiveness of the working of ADP, we at-first evaluated the two possible network learning strategies: Sensor-Initiated cluster formation and actuator-initiated cluster formation. The functioning of sensor initiated cluster formation is already defined in Section 3.1. So we look at the dynamics executed by the sensor and actuator nodes, in order to effectively finish the local-cluster using actor-initiated cluster formation approach. For the moment, we consider a random topology with 3 sensor nodes and only one actuator node, where any node has at-least one neighbor in its transmission range, in order to avoid freely hanging nodes in the network. Which in any case would be a useless assumption. Lets now explain how an actuator-initiated cluster formation would work. The actuator broadcasts an advertisement of its identity "ActuatorAdvertisement" in its transmission range. Upon receiving the advertisements, a sensor decides the optimal actuator if it has received multiple actuator advertisements, it sends an "JoinRrequest" packet to all the neighbors in its transmission range. Neighbors upon receiving the "JoinRequest" forward it to their neighbors until the probe arrives at the corresponding actuator. Actuator receives multiple requests by the same sensor node following different paths. In principle with most routing protocols, all the directed-broadcasts should be replied with some sort of acknowledgement, so the actuator replies with a "JoinAck" message. Now, lets assume that the actuator times out (to conserve sensors' energy consumption) for a certain time-period to receive all the "JoinRequest" message from one node and then it chooses the shortest path based on the available received information and then it sends an "JoinAck" to the sensor following this path. Finally when the sensor nodes receives the "JoinAck", it marks the forwarding neighbor as its one-hop neighbor for future transfer to-wards the actuator node.

If we carefully look at the steps taken by both strategies, there is not much visible difference. But, when a sensor node receives the actuator's broadcast, it sends a "JoinRequest" to its one-hop neighbors which reaches the actuator through multiple paths. Now this one-step neighbor broadcasting of "JoinRequest" is not at all similar to the one-step neighbor broadcasting of "AttachRequest" in our proposal (messages exchanged with ADP is shown by Figure 7(a). As shown in Figure 7(b), this leads to an immense increase in the overall number of packets sent and received before forming the local cluster. This effect gets worse with the increase in number of nodes in the network. And that certainly means an increased latency and more energy consumption during the network learning phase and is also taken care of before forming the proposal. One point to be noticed are the evolved local clusters which are more or less the same as obtained by the original proposition. By simply reverting the working of the ADP, decision of discovery-initiation process by actuators, increases the traffic load during the learning phase by 40-50% (unnecessary one-hop broadcast of “JoinRequest”), and hence the energy consumption and delay.



(a) Sensor-initiated Cluster Formation (b) Actuator-initiated Cluster Formation

Figure 7: Sensor-initiated Vs. Actuator-initiated Cluster Formation

5 Quantitative analysis of ADP

5.1 Experimental Method

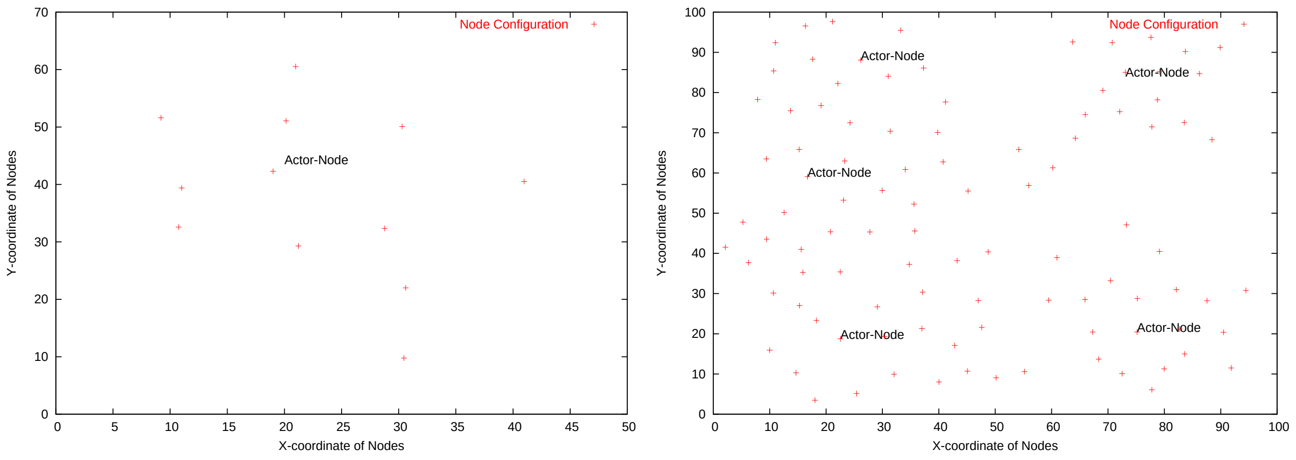
We have considered a wide range of network topologies and we test-validate the performance of ADP with all considered topologies. Due to the magnitude of work, we restricted ourselves to the static case, where both sensor and actuator nodes are static. In the light of all the above mentioned comparisons, we finally implemented the ADP in ns-2 [20] as an application layer protocol. At startup during the learning phase, the application starts the actuator-discovery. We present our results on the energy consumption and latency by the sensor nodes in different network scenarios. We also present the number of sensor nodes in each actuators' local cluster. In the end we obtain a delay CDF for different network considerations. Unlike prior work, We have considered a SANET as large as 95 sensor nodes and 5 actuator nodes, each node placed in coherence with the above two topologies. ADP is tailored to work with any routing protocol, but we will not discuss the routing protocol considerations for this work. Each actuator has sensor nodes at different hop counts in its local cluster. This gives a clear difference in the average-energy utilization by the sensors at different hop counts from the actuator node. An appropriate observation of energy consumption can lead to obtain a trade-off between the optimal number of sensor and actuator nodes for a wide range of sensor and actuator applications. We use the ns simulations to determine the effectiveness of ADP along with varying network conditions, that include:

Unstructured Network Topology: The unstructured network can be simple, comprising of a single actuator along with a very few sensor nodes or complex containing multiple actuator and sensor nodes. Each sensor node should have at-least one sensor node in its sensing range in order to avoid independently hanging nodes with no attached-actuator to forward their data. The reason behind naming this topology as unstructured comes from the fact that there is no attention given to the location of the nodes as the topology is generated randomly and we can have dense and scarce sensor node regions for a given topology. This consideration is tested only to obtain the worst delay and energy consumption by the sensor nodes in uncertain network topologies.

Structured Network Topology: This scenario can also have its simple and complex cases. But the maximum attention is paid to minimize the energy consumption and delay by simply manipulating the node positioning. If all the sensor nodes have the same sensing and transmission range, then the random network topology for performance evaluation looses its charm. And this is certainly done to also maximize the sensing range of the considered topology and to minimize the energy-consumption and latency.

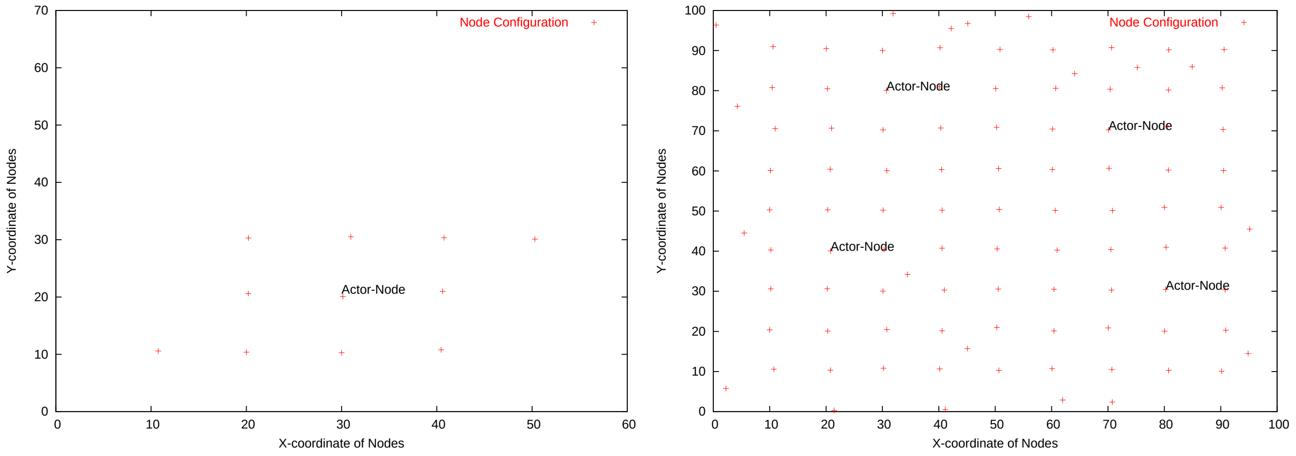
5.2 Configuration Topology

This section details the considered network topologies chosen to analyze the performance of ADP. We have actually conducted a huge series of simulation-runs to validate the effectiveness of ADP with both random and pre-determined topographical scenarios. Figure 8(a) shows a simple random network topology with 10 sensor nodes and only one actuator node. The positions of actuators are defined a priori at the start of the learning phase. Random topologies are chosen at the same time to ensure the working of ADP in any given network configuration. In Figure 9(a), the random topology considered earlier is optimized to evenly sense the deployment environment. The sensor nodes have predefined sensing and transmission radius, so the process of harmonizing the deployment yields in effective energy conservation and decreased delay. Following these possibilities, we can guarantee a certain promised QoS for most SANET applications ranging from dense employment monitoring to real-time applications. Similarly the Figure 8(b) shows a complex random network topology comprising of 95 sensor nodes and 5 actuator nodes. And Figure 9(b) shows the complex optimized network topology to evenly sense the environment. We have reduced the number of actuator nodes to 4 nodes during optimization for reasons we explain in a later section.



(a) Simple Topology (b) Complex Topology

Figure 8: Random Network Configuration

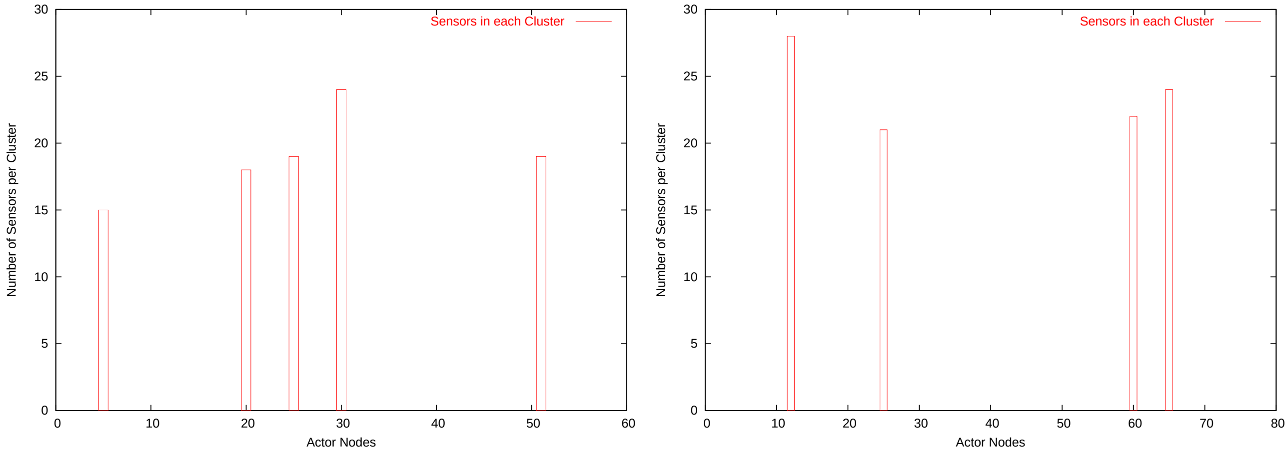


(a) Simple Topology (b) Complex Topology

Figure 9: Structured Network Configuration

5.3 Mean number of Sensors per Actuator

For the complex topologies with multiple sensors we have obtained graphs to show the average number of sensors per actuator in their local clusters. There is a natural balance among the number of sensors per actuator if the network is tailored to work effectively in terms of latency and energy consumption. Figure 10(a) and (b) show the average number of sensors per actuator (local cluster) for random and structured network topologies repectively. The local clusters in both cases (random and structured topology) have an equilibrium in terms of number of sensors, which proves the paths chosen by the sensors to their respective actuators are the shortest paths (Section 3.2). If we carefully look at the location of sensors and actuators in both network scenarios (Random and Structured), the sensors have chosen the closest geographical actuator (for an effective actuation process in the future). Which simply leads to efficient governance of global network objectives, such as compliance with time-stringent constraints and minimum energy consumption.



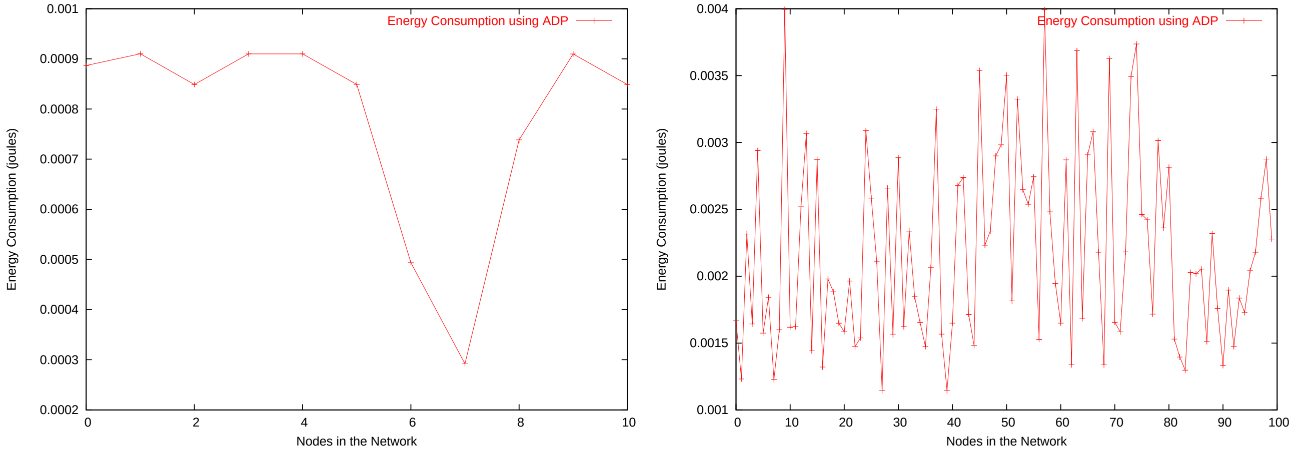
(a) Random Topology

(b) Structured Topology

Figure 10: Number of Sensors per Actuator for Complex Network Configuration

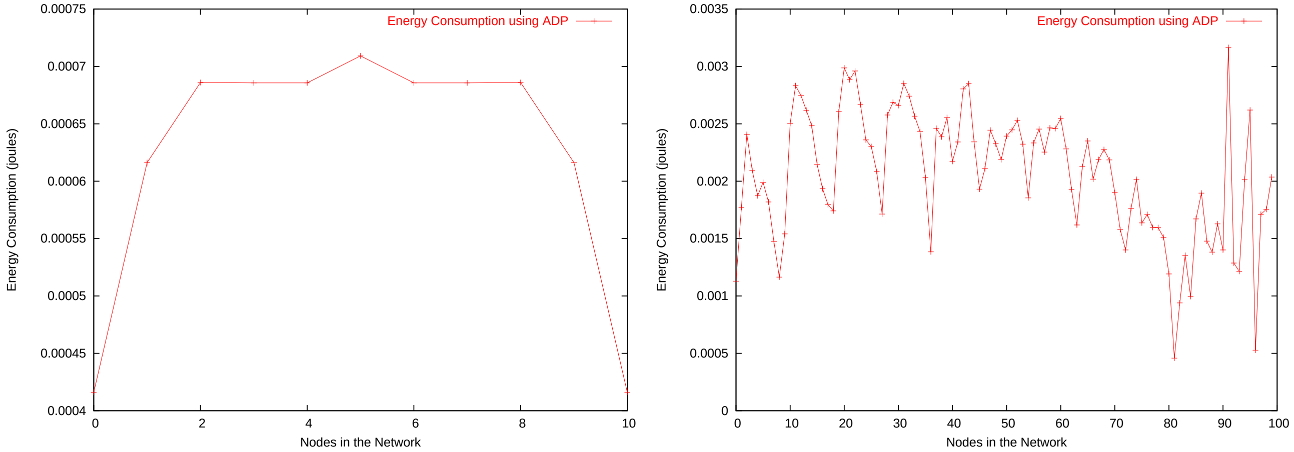
5.4 Energy Consumption

This section covers the explanation on energy consumption by each node during the actuator discovery phase. In Figure 11(a), the energy consumption by each sensor node to acquire the local cluster is presented. It depends on the amount of packets sent and received by any sensor node to finally become an integral part of a cluster. If we look at Figure 12(a), there is a sharp decrease in the energy consumption of the sensor nodes due to the presence of a symmetry in the deployment. In Figure 11(b), the individual sensor delays for a complex random network topology are presented. When we tried to un-randomize the topology and made it evenly sense the given environment, we find out that the average energy consumption of sensor nodes decreases even if we decrease the number of actuators by one as shown by Figure 12(b).



(a) Simple Topology (b) Complex Topology

Figure 11: Energy-Consumption of each Sensor for Random Network Configuration

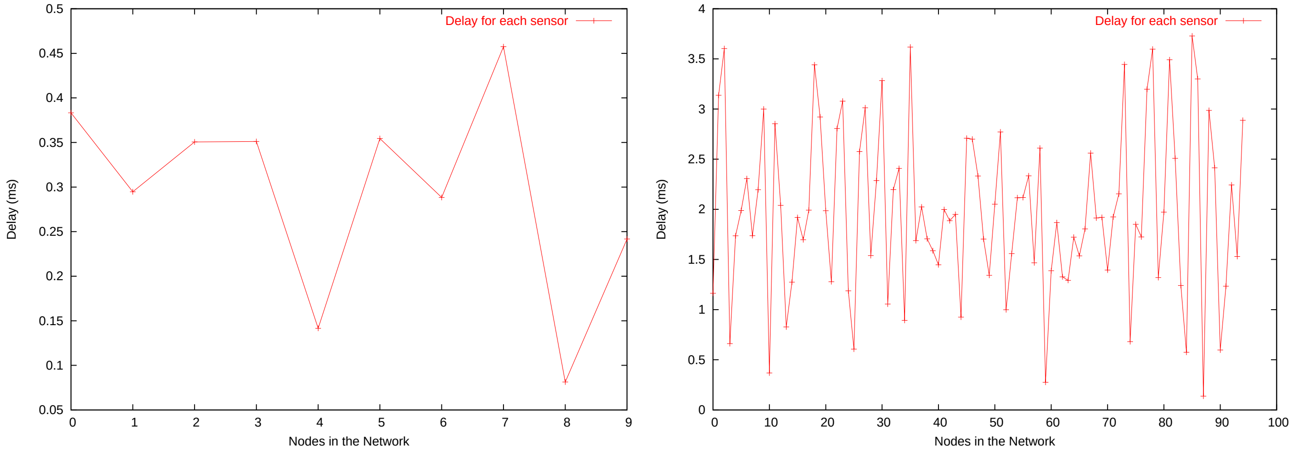


(a) Simple Topology (b) Complex Topology

Figure 12: Energy-Consumption of each Sensor for Structured Network Configuration

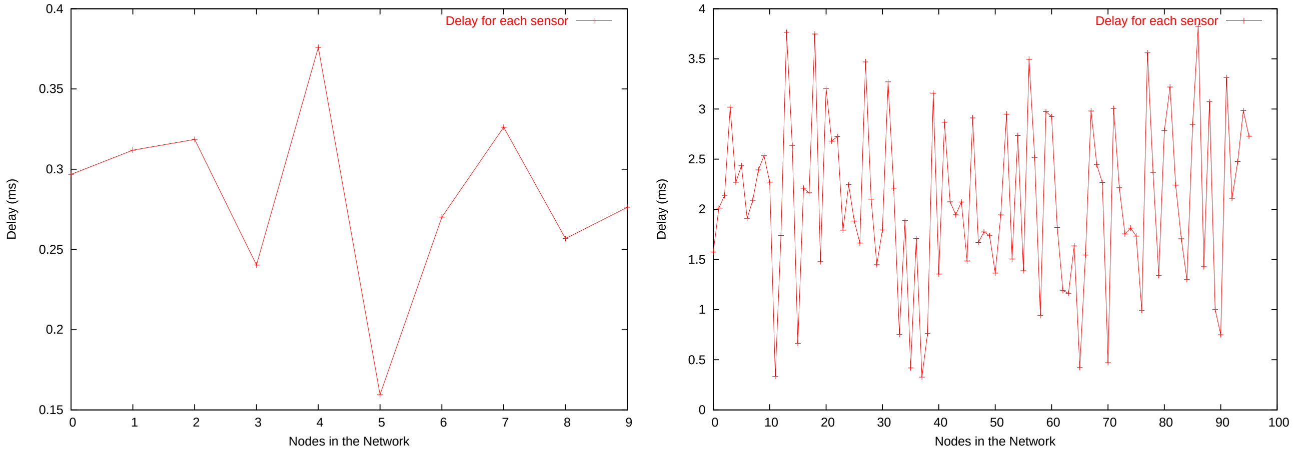
5.5 Observed Latency

This section briefly detail the individual delay (total time required to acquire a local cluster during the initial deployment) for each sensor node. The latency values shown in these graphs is the time between sending the “AttachRequest” and the reception of “JoinAck”. The delay for individual sensor nodes is presented in Figure 13(a) for our simple random network. There is a direct relationship between the energy utilization by any sensor node and the associated latency. The pursuit to optimize one results in two-fold benefits. Figure 14(a) justifies the improved delays for a structured network. This structuring phenomenon will eventually lead to an even sensing of the environment and effective actuation process. Figure 13(b) shows the delays obtained for the complex random network topology with 95 sensors and 5 actuator nodes. The difference in delay for the same number of sensor nodes with even fewer actuators (reduced to 4) clearly appears to be better as compared to the random case, shown in Figure 14(b) . As these paths need to be used in future, there is no further requirement of finding the routes whenever some sensed data needs to be transmitted to-wards the optimal actuators. For this end, we believe that even after including the MAC, queuing, and processing delay, the finally obtained latency values would still meet the real-time constraints for most SANET applications and satiating the energy-consumption constraints for densely-deployed sensor networks.



(a) Simple Topology (b) Complex Topology

Figure 13: The Attachment-Delay for each Sensor in Random Network Configuration

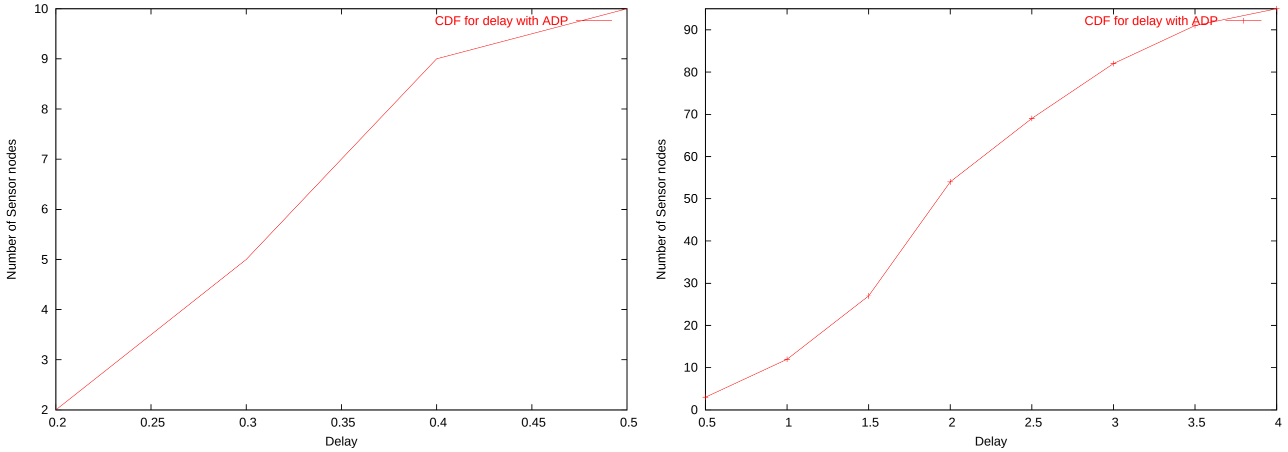


(a) Simple Topology

(b) Complex Topology

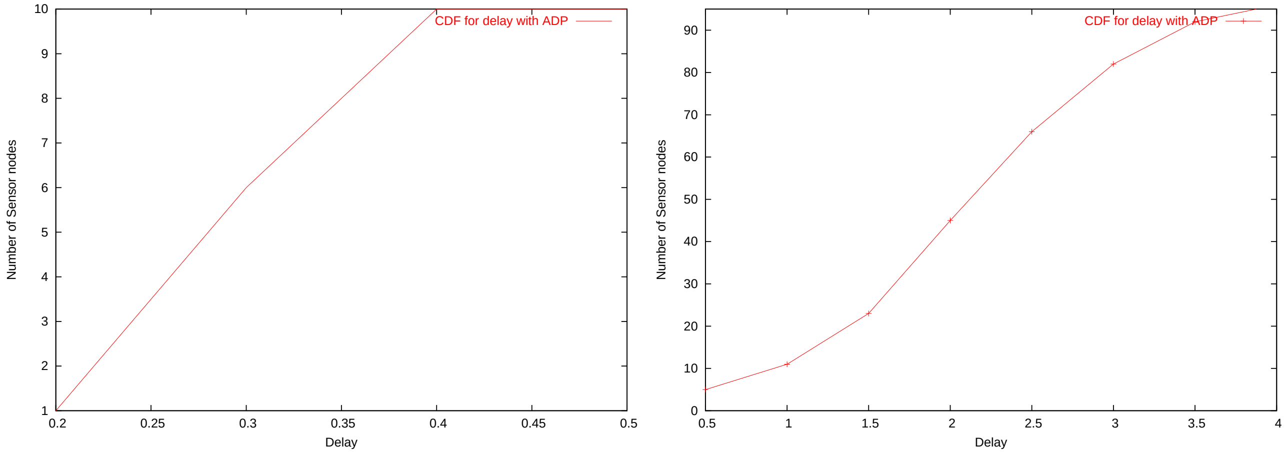
Figure 14: The Attachment-Delay for each Sensor in Structured Network Configuration

After obtaining the basic delay and energy consumption parameters, we tried to obtain the CDF for the delay distribution. According to our findings the delay for over 60% of the nodes lie in the average-delay (according to our findings) range as shown in Figure 15(a). Which is increased to more than 75%, Figure 16(a), in the case of evenly deployed sensor networks. The interest in topology control becomes interesting when the network grows complex as depicted by Figure 15(b) & Figure 16(b). The number of nodes in the average-delay zone increases just by obtaining an evenly scattered network (most suitable scenario for SANETs) deployment even when the number of actuator are decreased to 4 for the complex structured network configuration. These facts are evident from the slope of the graphs presented in this section.



(a) Simple Topology (b) Complex Topology

Figure 15: CDF for Attachment-Delay in Random Network Configuration



(a) Simple Topology (b) Complex Topology

Figure 16: CDF for Attachment-Delay in Structured Network Configuration

Along with the mean number of nodes per cluster we have also shown (in Figure 17) the mean path length as a function of network size. The mean path length, which is related to the end-to-end latency for both cases: Fixed number of actuators, and secondly actuators increasing by 5% with network size. However, the increase is more gradual with the increasing number of actuator nodes, which would be a more practical assumption for the SANETs.

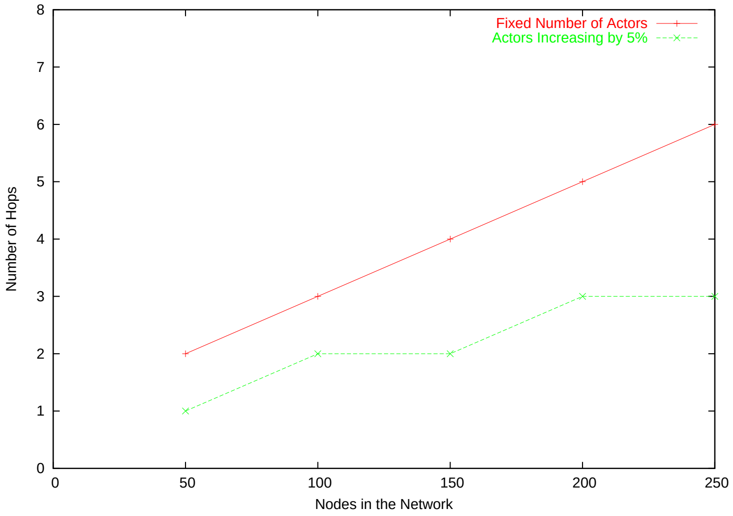


Figure 17: The Mean Path Length for ADP with Increasing Network Size

The actuators route their data to the sinks using AODV (Ad Hoc On-Demand Distance Vector routing) to confront with the real-time issues related to the validity of sensor data at the time of actuation. The motivation behind choosing AODV as a routing protocol for the actuator/actuator coordination is that the network of actuators could be seen as a Mobile Ad-Hoc Network (MANET) which in general has a low mobility. In this regard, we do-not present our results on actuator-actuator coordination for two concerns: The posed space problems and besides the AODV has already been shown to work effectively in all the considered scenarios.

6 Potential Applications with Design Aspects

At the current stage of research, there is no typical sensor/actuator network structure and the basic goals largely depend on the application. We have applied the right design approach to cope with a variable set of applications.

Deployment and Infrastructure: Sensor nodes can be deployed at specific locations or be place randomly. After initial deployment sensors may be added or replaced and the affected node locations along with the overall topology is well organized by the proposition.

Two common forms of communications modalities are infrastructure and ad-hoc networks. Both forms of support is available in our proposal with the additive advantage of seamless continuous communication among the nodes (both in case of mobility and node-failure) is supported by the proposal (for detail on design aspects, see [19]).

Network Topology and Size: Our multi-hop network can have any arbitrary graph with variable latency, robustness and capacity with a certain predefined and guaranteed QoS. The density μ is the number of the scattered nodes 'N' in the given region 'A' with 'R' as their radio transmission range [7].

$\mu = (\mathrm{N}\pi R^2)/\mathrm{A}$

This density function can be very useful in assuring the variable latency (min-max bound) and a guaranteed QoS during the deployment stage for both static and dynamic node-configurations as evident from the Figure 10(a) and (b).

Connectivity, Addressability and Lifetime: The communication ranges and physical locations of individual nodes define the connectivity of the network. There is always a network connection (over multiple hops) available between any two nodes, and the network can be considered as connected, even when the nodes are in sleep-mode the connection-path to them is still valid. In the case of mobility, if the path is no more available, an update request is sent immediately for an optimal path to the actuator node. The node addressability is application dependent but in our case, we have the leverage to address any two nodes to any depth in the network through a multi-hop path. Depending on the density, number and connectivity of the sensor nodes, we can approximately acquire a minimum bound on the maximum life time of the sensor network.

Query Ability, Data aggregation and Dissemination and real-time constraint: we have support for both types of node addressing; data-centric and node-centric. They both can be utilized interchangeably to create a balance between latency (lowest)and the energy-efficiency (maximum) depending on the application. An integrated support for data aggregation works flexibly well with all the major ways proposed for aggregation: diffusion algorithms, streaming queries, and event graphs. The three approaches differ in the way they influence the energy utilization and latency constraints, so it is left as an application and requirements specific concern to be monitored by the actuator nodes in the network. The ultimate goal of the sensor network is to detect specified event information of interest and deliver it to the actuator nodes. Where correlations can be utilized along the multi-hop path for efficient delivery semantics, which makes real-time an important concern for certain deployments.

Mobility and Self-Organization: Mobility has a large imp on the expected degree of network dynamics and hence influences the design of networking protocols and distributed algorithms. Theoretically we believed that our proposal works best with the mobility of nodes as compared to the previous routing protocols though it depends also on the actual speed of the mobile nodes, see [19] for details on mobility. The proposal takes into account the energy-efficiency of the nodes and the real-time constraints while searching a new path to the attached actuators in case of mobility. A node at any stage in its life is able to self-configure and to respond efficiently to any kind of fallacies that can occur, thus able to self-organize and adapt to the changing network conditions effectively.

7 Conclusions

For sensor-actuator coordination, the proposed ADP can guarantee ordering, synchronization and eliminates the redundancy of actions. For actuator-actuator coordination, the proposed unified framework using AODV can be exploited by different applications to always select the best networking paradigm that is possibly available according to the sensed information and to perform the necessary operations in an efficient way. As evident from the simulation results, ADP maintains a good control on energy-consumption by managing the average number of sensors per actuator (Local-Clusters) to almost an equilibrium. The energy utilization and delay constraints are significantly reduced as the result of optimal-path calculations during the initial deployment and using them till the network follows a topology change. Our results also showed that the latency and energy-consumption of about 60% of the sensor nodes falls in the average-case (if more sensor nodes fall in the average-consumption group, it will minimize the residual network-energy and forces the transmissions to-wards a low latency) for random network topology, and about 75% sensor nodes falls in the average-case for the structured network topology having 5% actuator population in the deployment. Localization and node positioning can be performed by improved MDS [9, 10] using actuator nodes as coordinates for effective energy utilization and fast localization of events for relevant actuation procedures. A major contribution to the state-of-the-art [9-19] in this paradigm can be directly applied to our proposal with minor modifications for a better utilization of the sensor nodes' energy and to cope with the real-time constraints posed by many applications. We are currently working on the implementation and performance analysis of mobile-configuration dynamics including: efficient failure-recovery and effectively eploiting the in-built coorelation (Spatial and Temporal) properties of SANETs.

References

[1] V. Paruchuri, A. Durresi, and L. Barolli, Energy Aware Routing Protocol for Heterogeneous Wireless Sensor Networks, in Proc. of Dexa, August 2005.

[2] Q. Fang, J. Gao, and L. J.Guibas, Locating and Bypassing Routing Holes in Sensor Networks, in Proc. of IEEE Infocom, March 2004.

[3] I.F. Akyildiz, I.H. Kasimoglu, Wireless sensor and actuator networks, research challenges. Article Ad Hoc Networks, Vol. 2, No. 4, pp. 351-367, October 2004.

[4] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: A survey, Article Computer Networks, Vol. 38, No. 4, pp. 393-422, March 2002.

[5] T. Melodia, D. Popili, V. C. Gungor and I. F. Akyildiz, A distributed coordination framework for wireless sensor and actuator networks, in Proc. of MobiHoc, May 2005.

[6] W. Hu, N. Bulusu, and S. Jha, A Communication Paradigm for Hybrid Sensor/actuator Networks, International Journal of Wireless Information Networks, Vol. 12, No. 1, pp. 47-59, January 2005.

[7] E. Yoneki, J. Bacon, A survey of WSN technologies: research trends and middleware's role, Technical Report, No. 646, September 2005.

[8] V. P. Mhatre, C. Rosenberg, D. Kofman, R. Mazumdar, N. Shroff, A Minimum Cost Heterogeneous Sensor Network with a Lifetime Constraint, IEEE Transactions on Mobile Computing, Vol. 4, No. 1, pp. 4-15, January 2005.

[9] Xiang Ji, and Hongyuan Zha, Sensor Positioning in Ad-hoc WSNs Using Multidimensional Scaling, in Proc. of IEEE Infocom, March 2004.

[10] Yi Shang, and Wheeler Ruml, Improved MDS-Based Localization, in Proc. of IEEE Infocom, March 2004.

[11] G. Sarma, and R. Mazumdar, Hybrid Sensor Networks: A Small World, in Proc. of MobiHoc, May 2005.

[12] A. Cerpa, J. L. Wong, M. Potkonjak, and D. Estrin, Temporal Properties of Low Power Wireless Links, in Proc. of MobiHoc, May 2005.

[13] I. F. Akyildiz, M. C. Vuran and O. B. Akan, On Exploiting Spatial and Temporal Correlation in Wireless Sensor Networks, in Proc. of IWWAN, May-June 2004.

[14] H. Gupta, V. Navda, S. R. Das, and V. Chowdhary, Efficient Gathering of Correlated Data in Sensor Networks, in Proc. of MobiHoc, May 2005.

[15] G. Xing, C. Lu, Y. Zhang, Q. Huang, and R. Pless, Minimum Power Configuration in Wireless Sensor Networks, in Proc. of MobiHoc, May 2005.

[16] F. Kuhn, T. Moscibroda, and R. Wattenhofer, Initializing Newly Deployed Ad Hoc and Sensor Networks, in Proc. of MobiHoc, May 2004.

[17] J. L. Bredin, E. D. Demaine, M. T. Hajiaghayi, and D. Rus, Deploying Sensor Networks with Guaranteed Capacity and Fault Tolerance, in Proc. of MobiHoc, May 2005.

[18] B. Liu, P. Brass, O. Dousse, P. Nain, and D. Towsley, Mobility Improves Coverage of Sensor Networks, in Proc. of MobiHoc, May 2005.

[19] M. Farukh Munir and F. Filali, A Novel Self Organizing Framework for SANETs, to appear in Proc, of the 12th European Wireless Conference (EW'06), April 2006.

[20] The Network Simulator - ns (version 2.27). http://www.isi.edu/nsnam/ns/.