Mobile Agent Middleware for Autonomic Data Fusion in Wireless Sensor Networks

Aristides Mpitziopoulos, Damianos Gavalas, Charalampos Konstantopoulos and Grammati Pantziou

Abstract Mobile Agents (MAs) are referred to as autonomous application programs with the inherent ability to move from node to node towards a goal completion. In the context of Wireless Sensor Networks (WSNs), MAs may be used by network administrators in the process of combining data and knowledge from different sources aiming at maximizing the useful information content. MAs have been initially developed to replace the client/server model which exhibits many disadvantages, particularly in WSN environments (e.g. heavy bandwidth usage and excessive energy expenditure). The most promising advantages of MAs in WSN environments include decreased usage of the wireless spectrum (large volumes of raw sensory data are filtered at the source) and energy consumption, increased reliability due to their inherent support for disconnected operations, their ability of cloning themselves to enable parallel execution of similar tasks, etc. The main objective of this book chapter is to review and evaluate the most representative MA-based middleware proposals for autonomic data fusion tasks in WSNs and evaluate their relevant strengths and shortcomings. Although the chapter's focus is on autonomic data fusion tasks, other applications fields that may benefit from the MAs distributed computing paradigm are identified. Open research issues in this field are also discussed.

Aristides Mpitziopoulos
Dept of Cultural Technology and Communication, University of the Aegean Address of Institute, Lesvos, Greece e-mail: crmaris@aegean.gr

Damianos Gavalas
Dept of Cultural Technology and Communication, University of the Aegean Address of Institute, Lesvos, Greece e-mail: dgavalas@aegean.gr

Charalampos Konstantopoulos
Research Academic Computer Technology Institute, Patras, Greece e-mail: konstant@cti.gr

Grammati Pantziou
Department of Informatics, Technological Educational Institution of Athens, Athens, Greece e-mail: pantziou@teiath.gr

1 Introduction

Wireless Sensor Networks (WSNs) typically comprise hundreds or even thousands of sensor nodes (SNs). These nodes are often randomly deployed in a sensor field and form an infrastructure-less network. Each node has the capability to collect data and route it back to a Processing Element (PE) or sink via ad hoc connections, using neighbor nodes as relays. A sensor consists of five basic parts: sensing unit, central processing unit (CPU), storage unit, transceiver unit and power unit [1].

Unlike other type of networks, WSNs are subject to a set of resource constraints such as limited energy availability of SNs [12, 48], since in most cases there are only dependant to battery supply, limited network communication bandwidth and hardware-network heterogeneity. In a typical WSN, SNs are equipped with restricted computational power and limited amount of memory for data-signal processing and task scheduling. Furthermore, as SNs are usually deployed to hostile environments they are prone to failures and the replacement of failed SNs is practically impossible in case of large-scale WSNs or embedded sensors. Given the fact that WSNs can be used in security sensitive applications (e.g. battlefield surveillance, secure area monitoring and target detection [1]) the above-mentioned limitations represent critical challenges.

WSNs can be used in a variety of applications, including environment monitoring, automatic target detection and tracking, battlefield surveillance, remote sensing, global awareness, etc [1]. A significant percentage of the above applications require remote retrieval of sensor readings and are known to be data-intensive. An efficient method to reduce the volume of data communicated within a WSN is data fusion. In data fusion, readings from multiple SNs are combined and processed leading to more accurate data with significant smaller size, since redundant readings for neighbor SNs are filtered [61]. However suitable middleware solutions are required to utilize data fusion in WSNs. In this context, middleware is defined as a software layer, which functions as an interface between sensor nodes and the data fusion process [17]. Mobile Agent (MA) technology has been proposed as an efficient middleware approach to IP networks, but is also suitable for WSN environments enabling the development and deployment of autonomic data fusion applications to resource constrained SNs. MAs are referred to as lightweight autonomic software entities that execute distributed tasks assigned by the users of a WSN, such as data fusion tasks.

The remainder of this chapter is organized as follows: First, we define the concept of data fusion, the related concepts of collaborative processing and data aggregation and the most representative approaches to WSN data fusion. We then present representative WSN middleware approaches and provide a comprehensive introduction to MA technology. Then we discuss the suitability, constraints and application fields of MAs in the context of WSNs. The next sections comprise the core of this chapter: we classify and review the most representative MA-based middleware proposals for autonomic data fusion in WSNs and evaluate their relevant strengths and shortcomings. Finally we refer to open research issues in this field.

2 Data Fusion in WSN environments

Data fusion is referred to as the process of combining data and knowledge from different sources with the aim of maximizing the useful information content [61]. It improves reliability while offering the opportunity to minimize the data retained.

Multi-sensor data fusion represents an evolving technology dealing with the problem of how to fuse data from multiple SNs to enable a more accurate estimation of the environment [45]. Such an approach achieves significant energy savings when intermediate SNs take part in the data fusion process (aggregate responses to queries). Applications of data fusion cross a wide spectrum, including environment monitoring, automatic target detection and tracking, battlefield surveillance, remote sensing, global awareness, etc [1]. They are usually time-critical, cover a large geographical area and require reliable delivery of accurate information for their completion. Madden et al. in [37] discuss the implementation of five basic database aggregates, i.e. count, min, max, sum, and average, based on the Tiny OS platform [57] and demonstrate that such a generic approach for data aggregation[1]leads to significant power (energy) savings. Other related works [21, 23, 33, 34] aim at reducing the energy expended by SNs during the process of data fusion. Most energy-efficient proposals are based on the traditional client/server computing model to handle multi-sensor data fusion in WSNs [19, 24, 28]; in that model, each SN sends its sensory data to a back-end PE or sink. However, as advances in sensor technology and computer networking allow the deployment of large amount of smaller and cheaper sensors, huge volumes of data need to be processed in real-time.

To address the above problem a new concept, collaborative processing [67] has been introduced referring to cooperative data processing, where data is combined from multiple sources. This feature also differentiates a sensor network from traditional centralized sensing and signal processing systems where raw data is collected by SNs and then is routed, without prior processing, though the network to a central PE which carries out the whole processing. This client/server approach presents many problems such as large energy consumption due to the transmission of large amount of raw data. This is especially true for SNs around the central node which constantly should relay data from other SNs. Also, this approach consumes scarce bandwidth resources leading so to severe scalability problems specially when there is large number of SNs that should send data to the PE.

Collaborative processing takes advantage of the correlation inherent within the information of neighbor SNs. Since sensing regions are largely overlapping, data from neighboring SNs refer to the same source of information (e.g. an evolving phenomenon, a moving target etc.). Hence, an aggregation and/or fusion of the original sensory data is possible. This processing takes place as data pass through SNs and not at the edge of network in a powerful PE. This approach drastically reduces the communicated data and hence relieves the network from the huge amount of data

[1] Data aggregation is the process of refining data from multiple sensors into a summarization based on some rules or criteria. Examples of aggregation methods are statistical operations like the mean or the median.

that would otherwise should have been communicated and then collected to the PE. Some approaches utilizing collaborative processing according to Qi et al. [47] are:

1. The Information-Driven approach, which has initially developed for target tracking applications [7, 66]. This approach enables energy-efficient computing through selecting the next SN which most likely improves the tracking accuracy (based on information, cost and resource constraints). This application is built on directed diffusion [2] as the communication medium.
2. The Relation-Based approach [20] wherein the environment is sensed based on high-level description of the task and then instructs selected SNs to sense and communicate their sensory data.
3. The Mobile Agent approach which provides more stable performance and improved fault tolerance than the information-driven approach, however, at the expense of extra bandwidth, needed for the transmission of the MA(s) code. This approach represents the main focus of this chapter.

3 Middleware Approaches in WSNs

The term middleware refers to the software layer positioned between the operating system and sensor applications on the one hand and distributed applications that interact with legacy systems on the other hand. The primary objective of the middleware layer is to hide the underlying complexity of the network environment by isolating the application from protocol handling, memory management, network functionality and parallelism [17].

The resource constraints (i.e. energy, limited memory and processing power) of contemporary nodes' hardware represent a challenge for the design of middleware solutions that meet the specific requirements of WSNs. Elen et al. in [8] categorized sensor middleware schemes into three main categories: application management where the middleware among other tasks has to deploy the application over the air on the WSN (i.e. Agilla [13], [3] and Mat*e* [31]), data management where the middleware has to handle the data packets that flows though the network (i.e. Milan [22], TinyDb [38] and Dsware [32]), and network service management where the middleware should offer a set of network services to the 'running' applications (i.e. Impala [35]).

MAs represent a promising middleware approach and in fact have been already used in some middleware schemes [3, 13]. A WSN Mobile-agent Middleware System (WMMS) must provide a platform to support MAs to perform user-assigned

[2] Directed diffusion [2, 7, 54] is a novel network protocol built for information retrieval and data dissemination. Its main characteristic is that it is 'data-centric', namely routing is based on data aggregated in the SNs rather than traditional IP theme where end-to-end delivery method is used based on unique identifications. Data generated by SNs is named by attribute-value pairs. A node requests data by sending interests for named data. Data matching the interest is then 'drawn' down towards that SN (set up of gradients). Intermediate SNs can cache or transform data and may direct interests based on previously cached data.

tasks (in our case data fusion tasks) while enabling the application deployment in WSNs.

4 Brief introduction to Mobile Agent technology

MA technology represents a relatively recent trend in distributed computing, which answers the flexibility and scalability problems of centralized models. The term MA [42] refers to an autonomous program with the ability to move from host to host and act on behalf of users towards the completion of an assigned task. In addition they are able to interact with legacy systems. Recently MAs have been proposed for efficient data dissemination in sensor networks [4, 45, 46, 56, 64, 65]. In a traditional client/server-based computing scheme, data from multiple nodes is transferred to one destination. Because the bandwidth of a WSN is typically much lower from other types of networks (e.g. wired networks), the data traffic derived from remote interactions may soon exceed the network capacity of a WSN. This serious scalability problem may be sufficiently addressed through MAs. The key idea is to delegate multiple MAs to the SNs of a WSN; these MAs can perform local rather than remote interactions with legacy systems, apply intelligent filtering operations thereby eliminating redundancy in the transferred data and decreasing the network overhead associated with data transfer. This approach also implies more reasonable usage of the nodes' radio unit (i.e. their most energy consuming part), hence prolonged nodes and network lifetime.

A MA may be defined as an entity of four attributes (see Fig. 1) [46]:

- Identification: a number used to uniquely identify the MA in the format of 2-tuple (i:j), where i indicates the IP address of the dispatcher and j the serial number assigned to agents by the dispatcher.
- Data space: the agent's data buffer which carries the partially integrated results (this result should provide progressive accuracy as the agent migrates from node to node).
- Itinerary: the route of migration. Itinerary planning includes two main issues that must be addressed: (a) the selection of SNs that must be visited; (b) the route that the MA will follow to visit the selected SNs. Itinerary scheduling can be classified as dynamic, static or hybrid[3].
- Methods: the application logic (or execution code) carried with the agent.

[3] A dynamic itinerary is determined on the fly at each hop of the MA, while a static itinerary is computed at the PE prior to the MA migration. In a hybrid approach the SNs to be visited are selected by the PE but the visiting order is decided on the fly by the MA. Although improving the optimality of the MA's itinerary compared to hybrid and static approaches, the dynamic approach is more time-expensive (the 'find-next-node' function is executed on each migration step), consumes valuable sensor nodes energy resources and implies larger MA sizes (the more intelligence integrated within the agent, the larger its size).

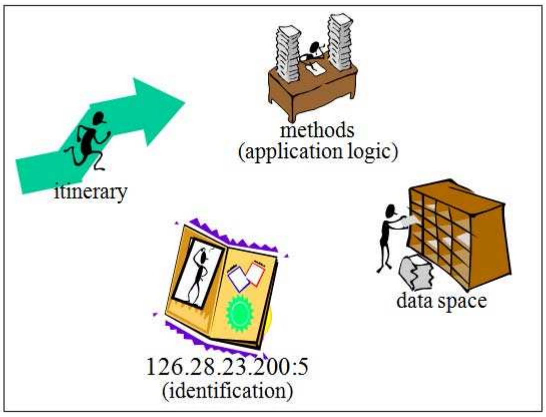

Fig. 1 Mobile agents as entities of four attributes.

5 Advantages/Disadvantages and Application Fields of MAs in the Context of WSNs

Lange and Oshima listed seven good reasons to use MAs [30]: reducing network load, overcoming network latency, robust and fault-tolerant performance, etc. The MA-based computing model enables moving the code (processing) to the data rather than transferring raw data to the processing module. By transmitting the computation engine instead of data, this model offers several important benefits:

- Decreased network overhead. Instead of passing large amounts of raw data over the network through several round trips, only an MA of small size is deployed. This is especially important for real-time applications and whenever the communication is enabled through low-bandwidth wireless connections. This also substantially implies improved network scalability.
- Distribution of network load. In the client/server model, the network area around the centralized element represents a bottleneck point (all remote calls are initiated and returned to this element). MAs favor network load balancing since agents are dispatched from the PE in the beginning of their journey and return back in the end of their itinerary. In the meanwhile, agent migrations typically occur in network regions away from the PE.
- Adaptability. MAs can be programmed to carry task-adaptive processes which may extend the built-in capability of the system on the fly.
- Stability and fault-tolerance. MAs can be dispatched when the network connection is alive and return results when the connection is re-established. Therefore, the performance of the system is not much affected by the reliability of network links.
- Autonomy. An agent is a self-contained software element responsible for performing part of a programmatic process. Therefore, it contains some level of

intelligence, ranging from simple predefined rules to self-learning Artificial Intelligence (AI) inference machines. It acts typically on behalf of a user or a process enabling task automation. MAs operate rather autonomously (they are often event or time triggered) and may communicate with the user, system resources and other MAs as required to perform their task. The autonomy feature of MAs mainly refers to their ability to exercise control over their own actions. That is, to respond in a timely fashion to changes in the environment, to exhibit goal-oriented behavior by taking the initiative and possibly change their itinerary on-the-fly depending on how they perceive their environment. In fact, autonomy has been agreed to comprise -among others- an essential feature of mobile agents [26, 29]. This feature along with platform and system independence makes them ideal for building robust and reliable WSNs. Furthermore MAs can support WSNs deployed in hostile environments, since they react dynamically to changes (for example interference or jamming).

Although the role of MAs in distributed computing is still being debated mainly because security concerns [15], several applications have shown clear evidence of benefiting from the use of MAs [40], including e-commerce and m-commerce trading [53], distributed information retrieval [25], network awareness [18], network & systems management [15, 49, 50], etc. Network-robust applications are also of great interest in military situations today. MAs are used to monitor and react instantly to the continuously changing network conditions and guarantee successful performance of the application tasks.

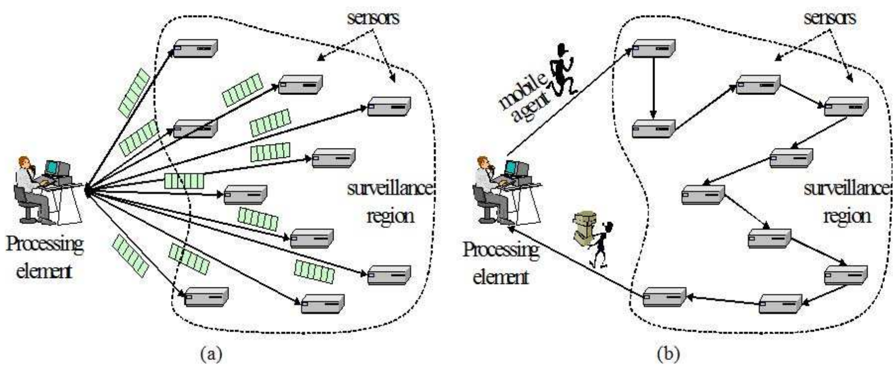

Fig. 2 Centralized vs. MA-based data fusion in WSNs.

MAs have also found a natural fit in the field of distributed sensor networks (see Fig. 2); hence, a significant amount of research has been dedicated in proposing ways for the efficient usage of MAs in the context of WSNs. In particular, MAs have been proposed for enabling dynamically reconfigurable WSNs through easy development of adaptive and application-specific software for SNs [58], for separating SNs in clusters [36], in multi-resolution data integration [56] and fusion [46], data dissemination [5] and location tracking of moving objects [2, 56, 63]. These applications involve the usage of multi-hop MAs visiting large numbers of SNs.

In MA-based autonomic data fusion tasks the choice of agents' itineraries is of critical importance affecting the overall energy consumption and data fusion cost. Notably, only few research articles have dealt with the problem of approximating optimal MA routes either through heuristics [45] or genetic algorithms [64]. The most notable weakness of these algorithms as argued in Sec. 7.1 is that they rely on a single MA to visit and fuse data from distributed sensors. However, such solutions do not scale acceptably for networks comprising hundreds or thousands of sensor nodes.

Security has been identified as the main reason that hinders the adoption of MAs as the next generation distributed computing paradigm[4]. In the context of WSNs, the most crucial security risk in using MAs is the possibility of tampering an agent. In a WSN that utilizes MAs the agent's code and internal data autonomously migrate between SNs and could be easy changed during the transmission or at a malicious (hostile) node. To address this security risk several countermeasures can be utilized to detect any manipulation made by an adversary, for instance Encrypted Functions (EF) [51], Cryptographic Traces [59, 60], Chained MAC protocol [11], Watermarking [10], Fingerprinting [10], Zero-knowledge proofs [43] and the Secure secret sharing scheme [43].

6 Classification of MA-Based Approaches for Autonomic Data Fusion in WSNs

MA-based approaches for autonomic data fusion in WSNs can be classified as follows (see Fig. 3):

- Single MA-based autonomic data fusion [5, 6, 45, 64], where only one MA is used for autonomic data fusion in the sensor network. This approach may work well with small scale WSNs but as discussed in Sec. 5 such solutions do not scale acceptably for networks comprising hundreds or thousands of sensor nodes.
- Multiple MA-based autonomic data fusion [16, 41, 46, 56], wherein a number of MAs is working in parallel to fuse data from WSN sensors. This approach is highly efficient even in large scale WSNs. However, it requires relatively complex algorithms to derive the itineraries of individual MAs.
- Autonomic data fusion in clustered WSN architectures [61]. This category refers to sensor networks with clustered structures. In such structures, SNs located in nearby locations are grouped in virtual clusters; one of these SNs acts as 'cluster

[4] The most critical security concerns related to MAs comprise: (a) protecting mobile hosts from malicious agents, (b) protecting agents from malicious hosts, and (c) protecting sensitive information carried by agents from eavesdropping. The first is addressed through implementing authentication and authorization features which ensure that only trusted agents may be executed, in a restricted authority domain. The second may be achieved through protecting agents against tampering, while the third is sufficiently addressed through encrypting sensitive data.

head' (CH) and sensory data retrieved by its 'cluster members' are rooted to the PE through the CH. Autonomic data fusion tasks within these clustered structures is assigned to MAs.

- Hardware-based autonomic data fusion [55]. In Sec. 4 we described MAs as autonomous software entities. Several research papers though used the term MA to refer to mobile hardware instances programmed with suitable software and acting as MAs (e.g. highly mobile SNs traversing the WSN to collect and process data from SNs and deliver it back to the PE).
- Combined multiple MA / stationary agents-based autonomic data fusion [39, 63]. This approach involves SNs with embedded Stationary Agents (SAs) that cooperate with their peers of neighbor nodes or MAs that traverse the network.

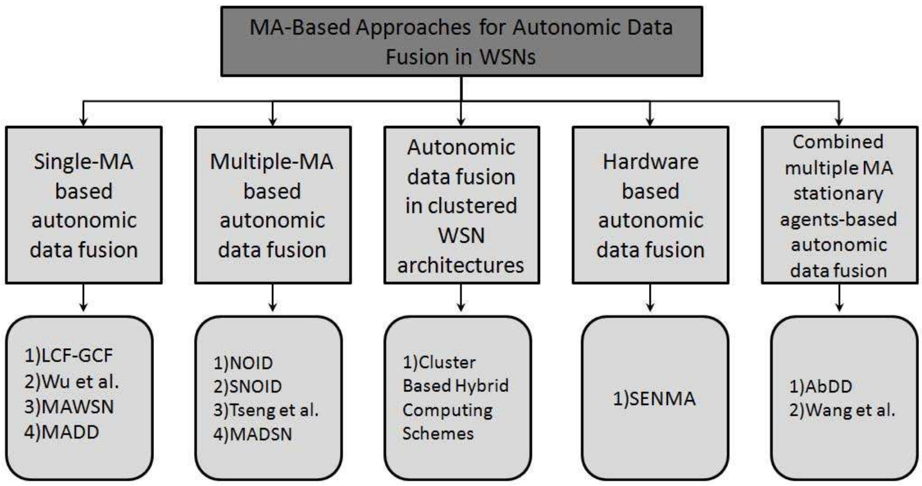

Fig. 3 A taxonomy of MA-based approaches for autonomic data fusion in WSNs.

7 MA-Based Approaches for Autonomic Data Fusion in WSNs

In this section we present the key concepts of the most representative MA-based autonomic data fusion schemes found in the literature, emphasizing on their relevant merits and shortcomings. The presentation is structured according to the classification provided in the preceding section (Sec. 6).

7.1 Single MA-Based Autonomic Data Fusion

7.1.1 Local Closest First and Global Closest First algorithms

Qi and Wang in [45] proposed two heuristics to optimize the itinerary of MAs involved in data fusion tasks. In Local Closest First (LCF) algorithm, each MA starts its route from the PE and searches for the next destination with the shortest distance to its current location. In Global Closest First (GCF) algorithm, MAs also start their itinerary from the PE node and select the node closest to the centre of the surveillance region as the next-hop destination.

The main asset of these two algorithms is that they are associated with low computational complexity. However, the output of LCF-like algorithms highly depends on the MAs original location, while the nodes left to be visited last are typically associated with high migration cost [27] (see, for instance, the last two hops, 6 and 7, in Fig. 4a); the reason for this is that they search for the next destination among the SNs adjacent to the MA's current location, instead of looking at the 'global' network distance matrix. On the other hand, GCF produces in most cases messier routes than LCF and repetitive MA oscillations around the region centre, resulting in long route paths and unacceptably poor performance [45, 64].

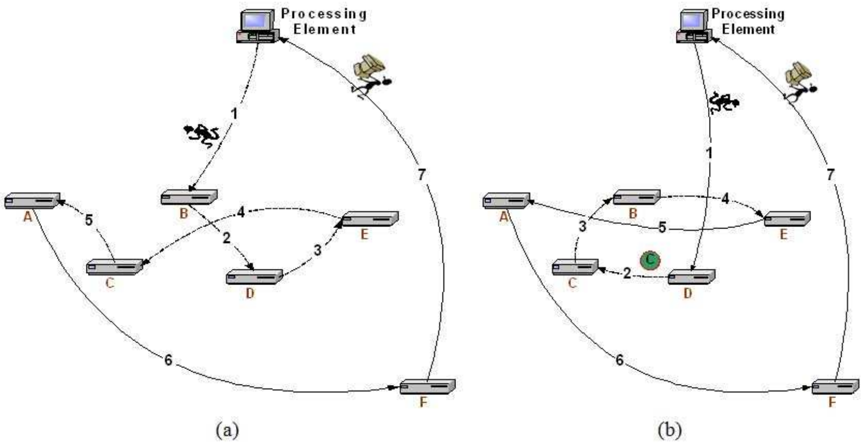

Fig. 4 (a) Output of LCF, (b) Output of GCF ('C' denotes the network's center).

The most serious drawback in both LCF and GCF is that they involve the use of a single MA object launched from the PE station that sequentially visits all SNs, regardless of their physical location on the plane. Their performance is satisfactory for small WSNs; however, it deteriorates as the network size grows and the sensor distributions become more complicated. This is because both the MA's round-trip delay and the overall migration cost increases squarely with network size, as the traveling MA accumulates into its state data from visited SNs [14]. The growing MA's state size not only results in increased consumption of the limited wireless bandwidth, but also consumes the limited energy supplies of SNs. This drawback is addressed by more recent works [16, 40] that propose methods for intelligent itinerary scheduling enabling the parallel employment of multiple MAs, each visiting a limited number of nodes.

7.1.2 Wu's et al. Genetic Algorithm

Wu et al. proposed a genetic algorithm [5] for computing routes for a MA that incrementally fuses the data as it visits the nodes in a WSN [64].The main application fields of this approach are object tracking and detection, where the MA must visit the sensors that sense the strongest signals, while the suggested itinerary must keep path loss and energy consumption low. The authors proved the above route computational problem to be NP-hard. Hence they relied on a genetic algorithm to solve the problem.

A two-level encoding is employed to adapt the genetic algorithm for the MA routing problem in WSNs. The first level is a numerical encoding of the sensor (ID) label in the order of SNs being visited by the MA. The second level is a binary encoding of the visit status of the SNs that are used in the first level (e.g. if a sensor has been visited, it is assigned '1' attribute value, else '0').

The proposed genetic algorithm is based on an event-driven adaptive method to implement a semi-dynamic routing strategy where the routing code is implemented exclusively in the PE. The MA carries only the pre-computed route that determines the order of SNs to be visited. In case of topology changes in the network (e.g. loss of nodes communication or energy depletion) that render the previous computed route invalid, the routing code is re-executed at the PE and the new route is sent to the MA.

Although providing superior performance (lower itinerary cost) than LCF and GCF algorithms, this approach implies a time-expensive optimal itinerary calculation (genetic algorithms typically start their execution with a random solution 'vector' which is improved as the execution progresses), which is unacceptable for time-critical applications, e.g. in target location and tracking. Also, in such applications, the group of visited SNs (i.e. those with maximum detected signal level) is frequently changed over time depending on target's movement; hence, a method that guarantees fast adaptation of MAs itineraries is needed.

7.1.3 Mobile Agent Based Wireless Sensor Network (MAWSN) architecture

Chen et al. in [5] proposed the Mobile Agent Based Wireless Sensor Network (MAWSN) architecture for filtering and aggregating data in planar sensor network architectures. In MAWSN, MAs are used to: (a) eliminate data redundancy among

[5] A genetic algorithm is a computational mechanism that 'simulates' the process of genetic selection and natural elimination in biological evolution.

SNs through applying context-aware local processing at the node level; (b) eliminate spatial redundancy among neighbor SNs by MA assisted data aggregation, since in WSNs comprising large numbers of SNs closely-located sensors generating redundant data are likely to exist; (c) reduce communication overhead by using a packet unification technique that concatenates the data from several short packets into one longer packet in order to reduce the communication overhead at the combined task level.

In MAWSN it is assumed that the PE is aware of the nodes that will be visited by the MA and the itinerary of the MA is predetermined. The payload of an MA is consisted of two parts, the processing code which is used to process sensed data and the aggregated data. Also the MA keeps a list (*SourceList*) with the source nodes that has to visit. In *SourceList*, there are two sources whose positions are important, namely, the first source which the MA will visit (*FirstSource*) and the last source (*LastSource*). The pair of *FirstSource* and *LastSource* represents the starting and ending points of the MA respectively, while *Nextsource* represents intermediate nodes.

When an MA is dispatched from the PE, it visits *FirstSource* where it is stored. *FirstSource* dispatches a copy of the stored MA (clone) after specific periods which are predefined according to the desired data rate. The clone after leaving *FirstSource* visits *Nextsource* nodes according to their gradient [54] (each time selects the one with maximum gradient), collects sensed data and deletes the current *NextSource* node from its *SourceList*. After visiting all *NextSource* nodes the clone finally reaches *LastSource*, where it also collects sensed data and then returns to the PE.

The authors proved via simulations that MAWSN presents performance gain over the client/server model in terms of energy consumption and packet deliver ratio. However, as the authors admit, MAWSN involve longer end-to-end latency under certain conditions due to the fact that only a single MA is employed in MAWSN. In scenarios wherein the MA visits large sets of sensors the latency (round trip delay of the MA) and the energy expenditure is drastically increased.

7.1.4 Mobile Agent Directed Diffusion (MADD) architecture

Chen et al. proposed the Mobile Agent Directed Diffusion (MADD) in [6] as an improvement to MA-Based Distributed Sensor Network (MADSN) and Multi-Resolution Integration (MRI) algorithm (see Sec. 7.2.4) The limitation of clustering in MASDN is addressed by using a flat network architecture (the authors argue that is more suitable for a wide range of WSNs applications compared to cluster-based architectures).

In MADD, MAs itineraries are scheduled by the PE utilizing directed diffusion [54]. The itinerary scheduling is the same with MAWSN with the only difference that when the clone MA reaches *LastSource* it discards the processing code and carries only the aggregated result to the PE saving valuable energy. The main differences between MADD and client/server model are:

- MADD uses a single MA that visits all the relevant SNs to collect data and the interval between the reports to the PE is decided by the dispatching rate of the MA. On the contrary in client/server-based WSNs the sensory data is sent individually by each sensor with a specified interval.
- In MADD, data is collected by the MA visiting all the SNs along a single path (itinerary), while in client/server-based WSNs, data is sent back to the PE in parallel from all nodes.

Although MADD addressed many constraints of MASDN, it failed to address the poor scalability of the approach wherein a single MA object visits sequentially the SNs for data collection. This renders both algorithms inappropriate for large-scale WSNs wherein the end-to-end delay and the size of the MA would increase squarely with the number of visited SNs.

7.2 Multi MA-Based Autonomic Data Fusion

7.2.1 The Near - Optimal Itinerary Design (NOID) Algorithm

Mpitziopoulos et al. in [41] proposed the Near-Optimal Itinerary Design (NOID) algorithm to address the problem of calculating a near-optimal route for a MA that incrementally fuses the data as it visits the nodes in a distributed sensor network.

NOID algorithm adapts some basic ideas of Esau-Williams (E-W) algorithm [9] and has been designed on the basis of three objectives: (a) MA itineraries should be derived as fast as possible and adapt quickly to changing networking conditions (hence, an efficient heuristic is needed), (b) MA itineraries should include only SNs with sufficient energy availability and exclude those with low energy level, (c) The number of MAs involved in the data fusion process should depend on the number and the physical location of the SNs to be visited; the order an MA visits its assigned nodes should be computed in such a way as to minimize the overall migration cost.

As opposed to single MA-based approaches, NOID enables the construction of multiple near-optimal itineraries, each assigned to individual MAs (see Fig. 5). NOID is executed on the PE platform; hence MA routes are predefined and not computed on-the-fly (awareness of the nodes' geographical locations is assumed). The authors claim this is a reasonable choice since MAs starts their journey from the PE node, which is typically equipped with powerful computing resources compared to SNs. However, a dynamic itinerary calculated at each hop by the MA would enable more prompt response to potential topology changes. On the other hand it would rise energy demands since the next-hop computation would execute on resource-constrained SNs; in addition, the MAs size would considerably increase (the itinerary scheduling logic would be embedded into MA code and transferred on every MA migration).

The authors also reported simulation results that demonstrated the improved NOID's performance over LCF and GCF algorithms in terms of the overall energy consumption and response time. This is mainly because NOID takes into account

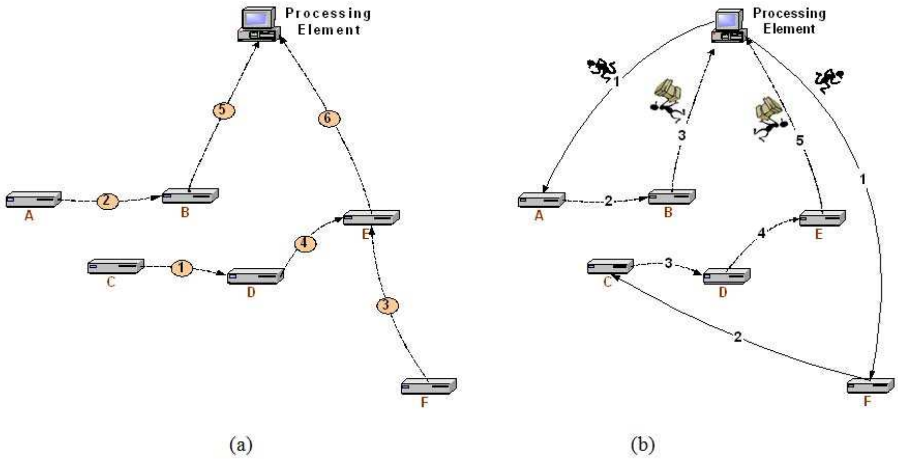

Fig. 5 (a) Output of NOID (the sequence numbers indicate the order in which the corresponding MA migrations are accepted, i.e., the algorithm's iteration sequence numbers), (b) the MA itineraries derived from the NOID algorithm's output.

the amount of data accumulated by MAs at each visited SN. Namely, NOID recognizes that traveling MAs become 'heavier' while visiting SNs without returning back to the PE to 'unload' their collected data [14]. Therefore, NOID promotes small itineraries enabling the parallel employment of multiple cooperating MAs, each visiting a subset of SNs.

7.2.2 The Second Near-Optimal Itinerary Design (SNOID) Algorithm

Gavalas et al. in [16] presented the Second Near-Optimal Itinerary Design (SNOID) algorithm for determining the number of MAs that should be used and the itineraries these MA should follow.

The main idea behind SNOID is to partition the area around the PE into concentric zones and start building the MA paths with direction from the inner (close to PE) zones to outer ones. The radius of the first zone which includes the PE is equal to $a \times r_{max}$ where a is an input parameter in the range $(0,1]$ and r_{max} is the maximum transmission range of any SN. All SNs inside the first zone are connected directly to the PE and these nodes are the starting points of the itineraries of the MAs. By adjusting the value of parameter a, the number of MA itineraries is adjusted accordingly.

Summarizing, SNOID algorithm determines the number of MAs by taking only into account the cost of communication between the PE and the first nodes of the itineraries. The remaining zones have a constant width equal to $r_{max}/2$. Thus, each SN can only directly communicate with nodes residing within the same zone or within the two adjacent zones.

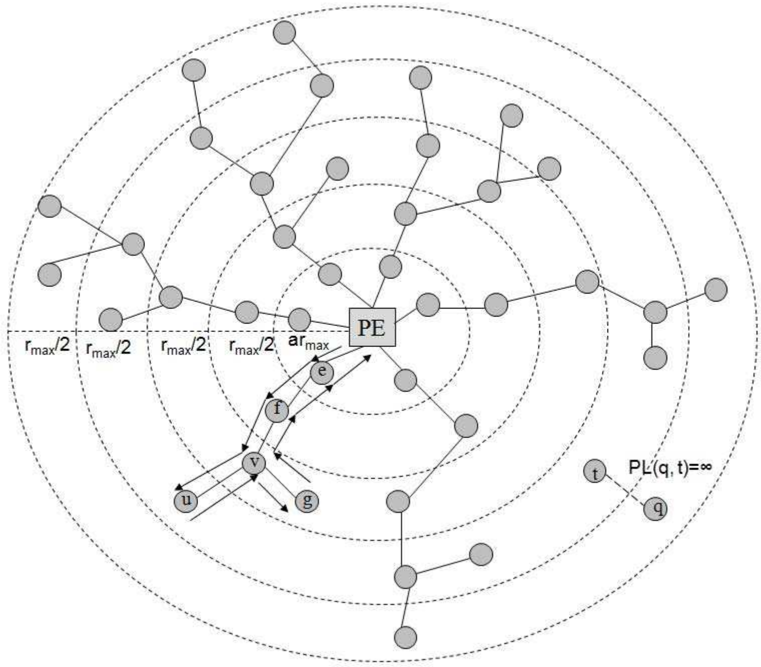

Fig. 6 Partitioning the area around PE into a number of zones (SNOID).

The construction of the itineraries of the MAs starts from the inner (close to PE) zones and proceeds to the outer zones, as illustrated in Fig. 6. When examining a zone, SNOID's objective is to connect each node with a node of the previous or the current zone which already has a connecting path back to the PE. Throughout this process attention is paid to the latency (denoted as PL in Fig. 6) of the trees formed up to that point, where latency is calculated through a simple formula. The nodes selected to join the trees are the ones that provide the minimum cost to the tree compared to other candidate nodes.

7.2.3 Location Tracking in a WSN by MAs and Its Data Fusion Strategies

In [56] Tseng et al. proposed the use of MAs for location tracking applications in WSNs to reduce sensing, computing and communication overheads. When a new object is detected by a SN, an MA is initiated to track the roaming path of the object. The MA visits the SN closest to the object and hops to other SNs following the object's movement. This MA (called master MA) may invite two nearby SNs to cooperatively position the object by dispatching a slave MA to each of them. Following that, the three MAs (the master MA and the two slave MAs) cooperate to perform the trilateration[6] algorithm [52] to calculate the object's precise location. As the object moves, the slave MAs may be revoked or reassigned depending on how 'strongly' they sense the moving object. Regarding the number of slave MAs the authors point out that although their development is based in the cooperation of only two, it will be straightforward to extend their work to more slave MAs to improve the positioning accuracy.

Besides location tracking the authors try to address the problem of fusion of data containing the tracking results. They propose two schemes to transfer the fused results to the PE, the Threshold-Based (TB) scheme and the Distance-Based (DB) scheme. In TB scheme the results are forwarded to the PE when the data size carried by the MA(s) reaches an upper bound while in DB scheme both data size and the distance of the MA from the PE are considered. Simulation results proved that DB performs well in all cases while in TB the threshold should be carefully chosen.

7.2.4 MA-Based Distributed Sensor Network (MADSN) and MRI Algorithm

To solve the problem of the overwhelming data traffic, Qi et al. [46] proposed the MA-based Distributed Sensor Network (MADSN) developed an enhanced version of the original multi-resolution integration (MRI) algorithm [44] for scalable and energy-efficient data aggregation. The idea of the original MRI algorithm [44] involves the construction of a simple function (overlap function) from the outputs of the sensors in a cluster and resolving this function at various successively finer scales of resolution to isolate the region over which the correct sensors lie. Each sensor in a cluster measures the same parameters. It is possible that some of them are faulty. Hence it is desirable to make use of this redundancy of the readings in the cluster to obtain a correct estimate of the monitored parameters.

The key concept in the enhanced version of MRI algorithm [46] for MADSNs is that multi-resolution analysis is applied at each sensor node instead of the PE, allowing MADSN to save up to 90% of data transfer time, according to the simulations the authors have conducted, compared to the original MRI [44] implementation for Distributed Sensor Networks (DSNs).

Concluding, in this approach by transmitting the software code (MA) to SNs that are nearby the area(s) of interest, a large amount of sensory data may be filtered at the source by eliminating the redundancy utilizing the enhanced MRI algorithm [46]. MAs may visit a number of SNs and progressively fuse retrieved sensory data, prior to returning back to the PE to deliver the data. This scheme may be more efficient than traditional client/server model; within the latter model, raw sensory data are transmitted to the PE where data fusion takes place.

The main drawback of MADSN is that it operates based on the following assumptions [6]: (1) the sensor network architecture is clustered; (2) source nodes are within one hop from a CH; (3) much redundancy exists among the sensory

[6] Trilateration is a method of determining the relative positions of objects using the geometry of triangles in a similar fashion as triangulation. It uses the known locations of two or more reference points, and the measured distance between the subject and each reference point.

data which can be fused into a single data packet with a fixed size. These assumptions pose many limitations on the range of applications that may be supported by MADSN. Also in [46] the authors assume a constant size for the MA, which is a non-valid assumption since MAs may get 'heavier' from the data they collect as they migrate through SNs.

7.3 Autonomic Data Fusion in Clustered WSN Architectures

7.3.1 Cluster-Based Hybrid Computing Schemes

Clustered architectures involve specific SNs operating as CHs; such nodes are typically positioned in their cluster's centre, while they act as relays for forwarding the sensory data retrieved by their assigned cluster members to the sink. Xu and Qi in [61] argued that mobile agent-based WSNs does not necessarily perform better than client/server schemes since MAs also introduce overhead due to their migrations and access to legacy systems resources.

Along this line, they proposed two cluster-based hybrid computing schemes that combine the advantages of MA and client/server models and offer better performance, should the proper scheme is chosen according to network clustering conditions. Within the first scheme (scheme A) each CH dispatches an MA that visits all the cluster members (SNs) in sequence to collect and aggregate data. When an MA returns to the CH, it sends the aggregated data back to the PE. In the second scheme (scheme B) an MA is dispatched by the PE and visits the CHs to collect the sensory data retrieved by the associated cluster members though client/server interactions.

The main drawback of the proposed hybrid model, as admitted by the authors, is that while for some network configurations it can make full use of the advantages of both the client/server and MA models, for some other configurations it could inherit the disadvantages of both models, leading to decreased performance. An example of such a situation is when the number of clusters is large for scheme A and small for scheme B. Also the authors in [61] assume a constant size for the MA, which is not realistic as the MA grows 'bigger' while collecting data from the SNs.

7.4 Hardware MA-based Autonomic Data Fusion

7.4.1 SENMA

Tong et al. in [55] proposed an architecture for large scale low-power sensor network, called Sensor Networks with Mobile Agents (SENMA) architecture. SENMA exploits node redundancies by introducing MAs that visit the SNs periodically or when the application requires for data gathering or network maintenance. The addition of MAs shifts computationally intensive tasks away from primitive SNs to more powerful MAs enabling energy efficient operations under severely limited power constraints.

MAs in SENMA are powerful hardware units equipped with sophisticated transceivers. Compared to regular SNs these special MAs are not strictly constrained on their communication and processing capability and their ability to traverse the sensor network. Examples of MAs could be manned/unmanned aerial vehicles, ground vehicles equipped with sophisticated terminals and power generators, or specially designed light nodes that can hop around in the network [55].

The authors showed that the simple topology of SENMA reduces energy consumption and improves the scalability of WSNs [55]. The main drawback of SENMA is the requirement of special hardware, playing the role of MAs that would significant increase the cost and the deployment complexity of the WSN. However, SENMA could be efficiently used in special applications where the cost of the WSN is not a first-priority issue (e.g. military applications).

7.5 Combined MA / Stationary Agents-Based Autonomic Data Fusion

7.5.1 Agent-Based Directed Diffusion (AbDD)

In [39] Agent-based Directed Diffusion (AbDD) is proposed by Malik et al. to address one of the most significant drawbacks in current routing schemes for WSNs: they all tend to propose optimal route that consume lowest energy (e.g. minimum number of hops path), leading all the nodes along the optimal path to faster energy depletion. On the contrary, AbDD ensures that data routing traffic is fairly balanced across the network as it takes into account both the routing cost and remaining energy of the nodes and utilizes the cloning capability of MAs. AbDD uses directed diffusion to address the redundancy of sensory data especially in large scale dense WSNs with arbitrary nodes placement. Apart from MAs, SAs permanently residing to SNs are also used (SAs directly interact with legacy systems, retrieve sensory data and report them to the MAs).

MAs and SAs in conjunction with directed diffusion achieve energy saving by moving the processing function to the data rather than transferring the data to the PE. AbDD's execution encompasses the following phases:

- Phase 1: The PE dispatches a single MA equipped with specified interest for data to identify -cooperating with the local SAs- the SNs that satisfy these interests (termed source SNs). In effect, an interest message is a query or an interrogation which specifies what the PE looks for. Each interest contains a description of sensing task that is supported by the WSN for acquiring data. When the MA locates a SN that meets the specified interests, it returns back to the PE building on the way the cost tables.

- Phase 2: The PE dispatches the MA to distribute the application-specific code to the source SNs. The sequence of source SNs to be visited is predetermined by the PE. However, MAs may alter their itinerary on-the-fly taking into account the battery level for each neighbor SN at each hop. When the MA reaches the first source SN, it distributes the copy of application code to the SA and clones itself to continue its itinerary. Next, it continues its itinerary and distributes copies of the application code at each source SN. When it reaches the last source SN is self-destroyed.
- Phase 3: The MA's clone residing in the first source SN makes a new clone that remains to the node (when a time interval set by the PE elapses). It then collects the data from the sensor and migrates to the other source SN performing data aggregation functions. Finally, it routes data back to the PE and destroys itself.

The authors proved through simulations that agent-based directed diffusion achieves lower energy comparison than distributed directed diffusion. However, although AbDD utilizes the cloning capability of MAs only a single MA initially is dispatched by the PE and a single MA visits the source SN and reports back to the PE. Thus, similarly to alternative single MA-based data fusion schemes AbDD is associated with increased energy consumption and latency when considering data retrieval from large sets of SNs.

7.5.2 Agent Collaborative Target Localization and Classification in Wireless Sensor Networks

Wang et al. in [63] proposed a heterogeneous architecture for WSNs, for target localization and classification tasks where both multi-agent systems (each node represents a multi-agent) and MAs are incorporated. The multi-agent system comprises a four-level hierarchy [63], where the top level is the interface agent. This agent is responsible for three procedures, namely receiving user queries about the environment, forwarding these queries to the lower level agents accordingly and reporting the query results to the users. In the immediately lower level resides the regional agent. According to the size of the WSN more than one regional agents may coexist, with each one being in charge of a region within the WSN. The task of the regional agents is to receive query requests from the interface agent and to coordinate sensor nodes within their region to collaboratively respond to the requests. A region is further split to clusters that are managed by manager agents. The role of manager agents is to directly control the behavior of sensor nodes (observing agents) that are given the task of sensing the current target-event.

In the above hierarchical multi-agent system, MAs are employed only when necessary and beneficial (e.g. in cases where data transmission comes in bulk or utilization of MAs gives superior performance). Sensor fusion is primarily performed based on multi-agent cooperation, however MAs are also used in cases of bulk data exchanges. This architecture greatly facilitates designs and implementations of WSN. In addition the architecture also readily adapts to diversified deployments at various scales.

Summarizing, the main purpose of this work is to develop the appropriate data fusion mechanisms, which should provide desirable accuracy and at the same time adapt to various WSN constraints (e.g. limited bandwidth and energy). They take advantage of both multi-agent and multi-MA schemes to achieve their goals.

8 Comparison of MA Approaches for Autonomic Data Fusion in WSNs

The performance of algorithms that fall into single-MA based category is relatively low especially in large scale WSNs. This is because a single MA is used for data fusion tasks and, hence, it must carry a heavy load of data retrieved from SNs. This category is only suitable for small-scale WSNs. On the other hand multi-MA based category can be efficiently used in large-scale WSNs, since multiple MAs are working in parallel for data fusion tasks leading to considerable energy gains. However, the complexity of the proposed algorithms is increased compared to single MA-based algorithms since they cater for scheduling non-overlapping itineraries for multiple MAs.

In clustered architectures the use of MAs for data fusion tasks may result in fast energy depletion of CH nodes. To overcome this limitation, clustering algorithms typically cater for the election of new CHs in periodic basis. However, that increases the complexity of MAs routing since the itinerary scheduling algorithm should be aware of the current CHs identity on every execution. Furthermore, this category has limited applicability to WSNs consisted of mobile SNs since their topology is frequently modified. Finally, CHs represent bottleneck points within the WSN, especially when transfer of large chunks of sensory data is involved.

The proposals in hardware-based category can offer comparable or improved performance than multi MA-based data fusion methods. Thus, they can efficiently be used in large scale WSNs where cost is not a prohibitive factor. Proposals that belong to combined MA / Stationary agents-based category can also be used in large scale WSNs because multiple MAs and SAs are utilized in data fusion process. These approaches offer comparable performance to that of multi agent-based models, however they imply complex manageability since they involve MA and SA entities. In addition, the permanent execution of SAs upon SNs implies increased demand upon device resources.

In Fig. 7 we summarize the advantages and disadvantages of all the research works reviewed in this chapter while Fig. 8 lists various parameters such as application field, number of utilized MAs and relevant category. We also refer to the parameters that affect the overall performance of each work (sensory data redundancy ratio and MA initial size applies to all approaches) and where the itinerary planning of MA(s) takes place.

	Pros	Cons
LCF - GCF	• Low complexity • Easy to implement	• Poor scalability: Increased energy consumption and latency in large scale WSNs
MAWSN	• Utilization of spatial redundancy • Reduction of communication overhead	
MADD	• Suitable for flat network architectures	
Wu et al.	• Superior performance compared to LCF-GCF	• Time-expensive itinerary computation (not adequate for time critical fusion tasks)
NOID	• Suitable for large scale WSNs (reduced communication overhead, decreased response time) • Takes into account nodes energy availability	• Increased complexity • The PE must know the position of each SN in the field
SNOID	• Suitable for large scale WSNs (reduced communication overhead, decreased response time) • Takes into account nodes communication range	
Tseng et al.	• Takes into account load of data carried by the MA • Takes into account spatial distance MA-PE	• Increased complexity
MADSN	• Suitable for large scale WSNs with clustered architecture	• Suitable only for clustered WSN architectures • Source nodes are assumed one hop away from a CH (one-hop clusters) • Assumes increased redundancy of sensory data
Cluster based hybrid computing schemes	• Uses the best computing model (among MA-based and client/server) depending on the network configuration • Inherits advantages of client/server and MA models	• Suitable only for clustered WSN architectures • In some WSN configurations it inherits the disadvantages of both client/server and MA models.
SENMA architecture	• Suitable for special WSN scenarios • Increased security of MAs	• Increased implementation cost • Need for specialized hardware
Abdd scheme	• Increased WSN lifetime	• Complex manageability (both MAs and SAs involved) • Increased demand upon device resources (SAs permanently reside on SNs)
Wang et al. scheme	• MAs are only used when needed reducing overhead • Scalable architecture	

Fig. 7 Advantages-Disadvantages of each proposed scheme.

9 Open Research Issues

The rapid advances in sensor technology depress the manufacturing cost of SNs and make feasible the deployment of large-scale WSNs. However the increased number of SNs implies increased data volumes that must be transferred through the limited bandwidth of wireless channels. This problem along with the resource constraints of contemporary SNs (e.g. limited energy, computation and communication capabilities) raises new challenges to the design of scalable and functional WSNs. To this end, the use of MAs for autonomic data fusion tasks has been a subject of intense research during the last few years. However, several research issues remain open, as outlined below:

- SNs in a WSN typically operate unattended, and are therefore vulnerable to tampering. Hence, the capture of an MA by an adversary is relatively easy and its collected data can then easily be retrieved. Also the MA may acquire deceitful data by a compromised node. Research should therefore be directed to MA-based schemes that provide effective, low-complexity security mechanisms for

	Application(s) field	Category	Parameters that affect the overall performance	Itinerary planning
LCF	data fusion	single-MA based	network scale	static
GFG	data fusion	single-MA based	network scale	static
Wu et al.	data fusion	single-MA based	network scale	hybrid
MAWSN	data fusion-aggregation	single-MA based	network scale	hybrid
MADD	data diffusion	single-MA based	network scale	static
NOID	data fusion	multiple-MA based	number of dispatched MAs	static
SNOID	data fusion	multiple-MA based	number of nodes within the PE range	static
Tseng et al.	location tracking data fusion	multiple-MA based	data size carried by the MA, spatial distance MA-PE	dynamic
MADSN	data fusion-aggregation	multiple-MA based	number of dispatched MAs	static
Cluster based hybrid computing schemes	data fusion	clustered WSN	number of clusters, average cluster size	static
SENMA architecture	data fusion-aggregation	hardware based	number of dispatched MAs	dynamic
Abdd	data fusion-aggregation data diffusion	combined multiple MA/SA based	network scale	hybrid
Wang et al.	location tracking data fusion	combined multiple MA/SA based	number of dispatched MAs number of clusters	dynamic

Fig. 8 Individual parameters of of each proposed scheme.

the MAs and privacy for their carried data. Techniques that allow MAs to identify tampered and unreliable SNs should also be investigated.

- WSNs can be threatened by Denial-of-Service (DoS) attacks (e.g. jamming, interference or resource exhaustion) which can cause the collapse of an entire network. DoS attacks represent a serious concern, especially in the eye of sensitive WSN application scenarios (e.g. battlefield surveillance). Thus, MA-based data fusion schemes able of responding to DoS attacks without disrupting their data fusion tasks should be derived.
- WSNs often face topology changes (due to temporal communication problems or nodes energy exhaustion), especially when deployed in hostile environments. Hence, research on MA-based data fusion should propose methods that guarantee the best attainable results in these environments. These methods should allow the fast and 'inexpensive' (of low complexity) adaptation of agent itineraries to topology modifications so that the overall fusion cost does not increase considerably. The key issues to be investigated is 'when' and 'how' to perform the itinerary adaptation; also 'who' (the PE or the MAs) will execute the itinerary adaptation procedure.
- Proposal of innovative WSN agent-oriented applications (apart from data fusion) that will benefit from the distributed nature of mobile agent objects and their ability to perform local data processing and filtering.
- Extensive evaluation of mobile agent paradigm in a variety of distributed sensor networks applications, such as object monitoring and tracking.
- Development of formal theoretical models that will systematically analyze the qualitative and quantitative trade-offs of the mobile agent approach vs the client-

server model in WSN environments, in a way similar to the one followed for IP networks [14].

10 Conclusion

This chapter reviewed the main aspects of data fusion, mobile agent technology and the benefits gained by utilizing MAs for autonomic data fusion tasks in WSNs.

It also classifies the research works that deal with MA-based autonomic data fusion in WSNs in five main categories: single MA-based, multiple MA-based, autonomic data fusion in clustered WSN architectures, hardware based and combined multiple MA / stationary agents-based autonomic data fusion, highlighting their relevant merits and shortcomings. Furthermore it highlights open research issues in the field of MA-based autonomic data fusion in WSNs. In the near future, the wider adoption and usage of WSN technologies is expected to bring out the significant role that MAs can play in this type of networks.

References

1. Akyildiz, F, Su W., Sankarasubramaniam Y., Cayirci, E.:A Survey on Sensor Networks, IEEE Communications Magazine, pp. 102-114, August 2002.
2. Boulis A.: Programming Sensor Networks with Mobile Agents, Proceedings of the 6th International Conference on Mobile Data Management (MDM'2005), pp. 252-256, May 2005.
3. Boulis, A., Han, C., Srivastava, M.:Design and Implementation of a Framework for Efficient and Programmable Sensor Networks, Proc. ACM MobiSys '03, pp. 187-200, May 2003.
4. Chong, C.Y., Kumar, S.P.:Sensor networks: evolution, opportunities, and challenges, Proceeding of the IEEE, Vol. 91, NO. 8, pp. 1247- 1256, Aug. 2003.
5. Chen, M., Kwon, T., Choi, Y.: Data Dissemination based on Mobile Agent in Wireless Sensor Networks, Proceedings of the 30^{th} IEEE Conference on Local Computer Networks (LCN'05), pp. 527-529, Nov. 2005.
6. Chen, M., Kwon, T., Yuan, Y., Choi, Y., Leug, V.C.M.: Mobile Agent-Based Directed Diffusion in Wireless Sensor Networks, EURASIP Journal on Advances in Signal Processing Volume 2007, Hindawi Publishing Corporation (2007).
7. Chu, M., Haussecker, H., Zhao, F.: Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks, International Journal of High Perfor-mance Computing Applications, (2002).
8. Elen, B., Michiels, S., Joosen, W., Verbaeten, P.: A middleware pattern to support complex sensor network applications, OOPSLA '06, Workshop on building software for sensor networks, Portland, Oregon, USA, (2006).
9. Esau, L.R., Williams K.C.: On teleprocessing system design. Part II- A method for approximating the optimal network, IBM Systems Journal, 5, pp.142-147 (1966).
10. Esparza, O., Fernandez, M., Soriano, M., Munoz, J.L., Forne, J.: Mobile Agents Watermarking and Fingerprinting: Tracing Malicious Hosts, DEXA 2003, LNCS 2736, Sproger-Verlag, (2003).
11. Fisher, L.: Protecting Integrity and Secrecy of Mobile Agents on Trusted and Non-Trusted Agent Places, diploma thesis, Department of computer science, University of Bremen, (2003).

12. Flinn, J., Satyanarayanan, M.: Energy-aware adaptation for mobile applications, Symposium on Operating Systems Principles, pp. 48-63, (1999).
13. Fok, C., Roman, G., Lu, C.: Mobile Agent Middleware for Sensor Networks: An Application Case Study, Proc. IEEE IPSN '05, pp. 382- 387, Apr. 2005.
14. Fuggeta, A., Picco, G.P., Vigna, G.: Understanding Code Mobility, IEEE Transactions on Software Engineering 24(5), pp. 346-361, (1998).
15. Gavalas, D.: Mobile Software Agents for Network Monitoring and Performance Management, PhD Thesis, University of Essex, UK, July 2001.
16. Gavalas, D., Pantziou, G., Konstantopoulos, C., Mamalis, B.: New Techniques for Incremental Data Fusion in Distributed Sensor Networks, In Proceedings of the 11th Panhellenic Conference on Informatics (PCI'2007), pp. 599-608, (2007).
17. Geihs, K.: Middleware Challenges Ahead, IEEE Computer, pp. 24-31, (2001).
18. Al-Hammouri, A., Zhang, W., Buchheit, R., Liberatore, V., Chrysanthis, P., Pruhs, K.: Network Awareness and Application Adaptability, Information Systems and E-Business Management, 4(4), pp. 399-419, Oct. 2006.
19. Iyengar, S. S., Jayasimha, D. N., Nadig, D.: A versatile architecture for the distributed sensor integration problem, IEEE Trans. Comput., 43(2), pp. 175-185, Feb. 1994.
20. Guibas, L. J. Sensing, tracking, and reasoning with relations. IEEE Signal Processing Magazine 2002, pp. 73-85, (2002).
21. Heinzelman, W., Kulik, J., Balakrishnan, H.: Adaptive Protocols for Information Dissemination in Wireless Sensor Networks, in Proceedings of 5^{th} ACM/IEEE Mobicom Conference, pp. 174-185, (1999).
22. Heinzelman, W. B., Murphy, A. L., Carvalho, H. S., Perillo M A.: Middleware to Support Sensor Network Applications, IEEE Network, pp. 6-14, January/February 2004.
23. Intanagonwiwat, C., Govindan, R., Estrin, D.: Directed diffusion: a scalable and robust communication paradigm for sensor networks, Mobile Computing and Networking, pp. 56-67, (2000).
24. Jayasimha, D. N., Iyengar, S. S., Kashyap, R. L.: Information integration and synchronization in distributed sensor networks, IEEE Trans. Syst., Man, Cybern., 21(21) , pp. 1032-1043, Sept./Oct. 1991.
25. Jiao,Y., Hurson, A.R.: Adaptive Power Management for Mobile Agent-Based Information Retrieval, Proceedings of the 19^{th}19 International Conference on Advanced Information Networking and Applications (AINA'05), pp. 675-680, March 2005.
26. Kendall, E. A., Krishna, P. V. M., Pathak, C.V., Suresh, C. B.:Patterns of Intelligent and Mobile Agents, Proceedings of the 2nd International Conference on Autonomous Agents (Agents98), pp. 92-99, May 1998.
27. Kershenbaum, A.: Telecommunications Network Design Algorithms, McGraw-Hill, (1993).
28. Knoll, A. and Meinkoehn, J.: Data fusion using large multi-agent networks: an analysis of network structure and performance, in Proceedings of the International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Las Vegas, NV, IEEE, pp. 113-120, Oct. 2-5 1994.
29. Krause, S., Magedanz, T.: Mobile Service Agents enabling Intelligence on Demand in Telecommunications, Proceedings of IEEE GLOBCOM 96, pp. 78-84 , November 1996.
30. Lange, D. B., Oshima, M.: Seven Good Reasons for Mobile Agents, Communications of the ACM, 42(3), pp. 88-89, March 1999.
31. Levis, P., Culler, D.: Mat?: a tiny virtual machine for sensor networks, In ASPLOS- X: Proceedings of the 10th international conference on Architectural support for programming languages and operating systems, pp. 85-95, (2002).
32. Li, S., Son, S. H., Stankovic, J. A.: Event detection services using data service middleware in distributed sensor networks, in IPSN'03:Proceedings of the 2nd international symposium on Information processing in sensor networks, pp. 502-517, (2003).
33. Lindsey, S., Raghavendra, C. S.: PEGASIS: Power Efficient GAthering in Sensor Information Systems. In Proceedings of IEEE Aerospace Conference, vol.3, pp. 1125-1130, (2002).

34. Lindsey, S., Raghavendra, C. S., Sivalingam, K.:Data Gathering in Sensor Networks using the Energy*Delay Metric. In Proceedings of the IPDPS Workshop on Issues in Wireless Networks and Mobile Computing, pp. 2001-2008, (2001).
35. Liu, T., Martonosi, M.: Impala: a middleware system for managing autonomic, parallel sensor systems, In PPoPP'03: Proceedings of the ninth ACM SIGPLAN symposium on Principles and practice of parallel programming, pp.107-118, (2003).
36. Lotfinezhad, M., Liang, B.: Energy Efficient Clustering in Sensor Networks with Mobile Agents. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'05), vol. 3, pp. 1872-1877, Mar. 2005.
37. Madden, S., Franklin, M.J., Hellerstein, J.M.: TAG: A Tiny Aggregation Service for Ad-Hoc Sensor Networks. In Proceedings of the 5th Annual Symposium on Operating Systems Design and Implementation (OSDI'02), vol.36, pp. 131-146, USENIX, Dec. 2002.
38. Madden, S., Franklin, M. J., Hellerstein, J. M., Hong, W.: The design of an acquisitional query processor for sensor networks, In SIGMOD'03: proceedings of the ACM SIGMOD international conference on management of data, pp. 491-502, (2003).
39. Malik, H., Shakshuki, E., Dewolf, T.: Multi-agent System for Directed Diffusion in Wireless Sensor Networks, 21st International Conference on Advanced Information Networking and Applications Workshops, vol. 2, pp. 635-640, Niagara Falls, ON, Canada, May 2007.
40. Milojicic D.: Mobile Agent Applications, IEEE Concurrency, 7(3), July-Sep. 1999.
41. Mpitziopoulos, A., Gavalas, D., Konstantopoulos, C., Pantziou, G.:Deriving efficient mobile agent routes in wireless sensor networks with NOID algorithm. In Proceedings of the 18th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'2007), Athens, Greece, pp. 1-5, September 2007.
42. Pham, V., Karmouch, A.: Mobile Software Agents: An Overview, IEEE Communica-tions Magazine, 36(7), pp. 26-37, (1998).
43. Pieprzyk, J., Hardjono, T., Seberry, J.: Fundamentals of Computer Security, Springer-Verlag, Berlin (2003).
44. Prasad, L., Iyengar, S. S., Rao, R. L.:Fault-tolerant sensor integration using multiresolution decomposition, Physical Review E, vol. 49, no. 4, pp. 3452-3461, April 1994.
45. Qi, H., Wang, F.: Optimal Itinerary Analysis for Mobile Agents in Ad Hoc Wireless Sensor Networks, Proceedings of the13th International Conference on Wireless Communica-tions (Wireless'2001), pp. 147-153, (2001).
46. Qi, H., Iyengar, S. S., Chakrabarty, K.: Multi-resolution data integration using mobile agents in distributed sensor networks, IEEE Trans. on Syst., Man, and Cybern. Part C: Applications and Reviews 31 (3), pp. 383-391, (2001).
47. Qi, H., Kurganti, P. T., Xu, Y.: The Development of Localized Algorithms in Wireless Sensor Networks, Sensors (2002), 2, pp. 286-293, (2002).
48. Raghunathan, V., Schurgers, C., Park, S., Srivastava, M. B.: Energy-aware wireless microsensor networks, IEEE Signal Processing Magazine 19 (2) pp. 40-50,(2002).
49. Reuter, E., Baude, F.: System and Network Management Itineraries for Mobile Agents. Proceedings of the 4th International Workshop on Mobile Agents for Telecommunication Applications (MATA'02), LNCS vol. 2521, pp. 227-238, October 2002.
50. Rubinstein, M.G., Duarte, O. C., Pujolle, G.: Scalability of a Mobile Agents Based Net-work Management Application, Journal of Communications and Networks, 5(3), September 2003.
51. Sander, T., Tschudin, Ch. F.: Protecting mobile agents against malicious hosts. In Mobile Agents and Security, volume 1419 of LNCS. Springer-Verlag, 1998.
52. Savvides, A., Han, C.C., Srivastava, M.B.: Dynamic fine-grained localization in ad-hoc networks of sensors, in Proc. Seventh Ann. ACM/IEEE Int. Conf. on Mobile Computing and Networking (Mobicom 2001), Rome, Italy, pp. 166-179, ACM press, (2001).
53. Shih, D.H., Huang, S.Y., Yen, D.C.: A New Reverse Auction Agent System for m-Commerce Using Mobile Agents, Computer Standards & Interfaces, 27(4), pp. 383-395, April 2005.
54. Silva, F., Heidemann, J.,Govindan, R.,Estrin, D.: Directed Diffusion, Technical Report ISI-TR-2004-586, USC/Information Sciences Institute, January, (2004).
55. Tong, L., Zhao, Q., Adireddy, S.: Sensor Networks with Mobile Agents,, in Proc. IEEE MILCOM'03, pp. 1-6, Boston, MA, Oct. 2003.

56. Tseng, Y.C., Kuo, S.P., Lee, H.W., Huang, C.F.: Location Tracking in a Wireless Sensor Network by Mobile Agents and Its Data Fusion Strategies, Computer Journal, 47(4), pp. 448-460, (2004).
57. Tiny Os: http://www.tinyos.net/. Cited 20 Feb 2008.
58. Umezawa, U., Satoh, I., Anzai, Y.: A Mobile Agent-Based Framework for Configurable Sensor Networks, Proceedings of the 4th International Workshop on Mobile Agents for Telecommunications Applications (MATA'02), pp. 128-140, Oct. 2002.
59. Vigna, G.: Protecting Mobile Agents through Tracing, Proceedings of the 3rd ECOOP Workshop on Mobile Object Systems, Finland, June 1997.
60. Vigna, G.: Cryptographic traces for mobile agents". In Mobile Agents and Security, vol 1419 of LNCS. Springer-Verlag, (1998).
61. Xu, Y., Qi, H.: Distributed Computing Paradigms for Collaborative Signal and Information Processing in Sensor Networks, Int'l. J. Parallel and Distrib. Comp., 64(8), pp. 945-959, Aug. 2004.
62. Wald L.: Some terms of reference in data fusion. IEEE Transactions on Geosciences and Remote Sensing, 37, 3, pp. 1190-1193, (1999).
63. Wang, X., Bi, D.W., Ding, L., Wang, S.: Agent Collaborative Target Localization and Classification in Wireless Sensor Networks. Sensors 2007, 7, pp. 1359-1387, (2007).
64. Wu, Q., Rao, N., Barhen, J., Iyengar, S., Vaishnavi, V., Qi, H., Chakrabarty, K.: On Computing Mobile Agent Routes for Data Fusion in Distributed Sensor Networks. IEEE Transactions on Knowledge and Data Engineering, 16(6), pp. 740-753, June 2004.
65. Zaslavsky, A.:Mobile Agents: Can They Assist with Context Awareness?, 2004 IEEE MDM'04, pp. 304-305, Berkeley, California, Jan. 2004.
66. Zhao, F., Shin, J., Reich, J.: Information-driven dynamic sensor collaboration for tracking applications, IEEE Signal Processing Magazine, vol. 19(2), pp. 61-72, Mar. 2002.
67. Zhao, F., Guibas, L.: Wireless Sensor Networks, Morgan Kaufmann Publishers, San Francisco, (2004).