Computer Networks 54 (2010) 2991–3010

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Ant-based routing for wireless multimedia sensor networks using multiple QoS metrics

Luis Cobo *, Alejandro Quintero, Samuel Pierre

Mobile Computing and Networking Research Laboratory (LARIM), Department of Computer Engineering, École Polytechnique de Montréal, Quebec, Canada H3T 1J4

ARTICLE INFO

Article history:
Received 1 June 2009
Received in revised form 10 May 2010
Accepted 20 May 2010
Available online 31 May 2010
Responsible Editor: T. Melodia

Keywords:
Quality of service
Routing protocols
Wireless multimedia sensor networks
Ant routing algorithm
Video distortion

ABSTRACT

In wireless sensor networks, most routing protocols consider energy savings as the main objective and assume data traffic with unconstrained delivery requirements to be a given. However, the introduction of video and imaging sensors unveils additional challenges. The transmission of video and imaging data requires both energy efficiency and QoS assurance (end-to-end delay and packet loss requirements), in order to ensure the efficient use of sensor resources as well as the integrity of the information collected. This paper presents a QoS routing model for Wireless Multimedia Sensor Networks (WMSN). Moreover, based on the traditional ant-based algorithm, an ant-based multi-QoS routing metric (AntSensNet) is proposed. The AntSensNet protocol builds a hierarchical structure on the network before choosing suitable paths to meet various QoS requirements from different kinds of traffic, thus maximizing network utilization, while improving its performance. In addition, AntSensNet is able to use a efficient multi-path video packet scheduling in order to get minimum video distortion transmission. Finally, extensive simulations are conducted to assess the effectiveness of this novel solution and a detailed discussion regarding the effects of different system parameters is provided. Compared to typical routing algorithms in sensor networks and the traditional ant-based algorithm, this new algorithm has better convergence and provides significantly better QoS for multiple types of services in wireless multimedia sensor networks.

1. Introduction

Rapid advances in Micro-Electro-Mechanical System (MEMS) technology, proliferation of wireless communication and digital electronics have set the stage for the deployment of low-cost, low-power, multi-functional, autonomous sensor networks. The major objectives behind the research and deployment of Wireless Sensor Networks (WSNs) [1] lie in the following two broad aspects: (i) event detection (sensing) and data communication through node coordination and (ii) conservation of energy to maximize the post-deployment, active lifetime of individual sensor nodes and the overall network. On the other hand, today's wireless communication is gradually changing the paradigms from the existing scalar services (light, temperature, etc.) to a new world of real-time audio-visual applications. The increasing popularity of multimedia applications has already given birth to the new term Wireless Multimedia Sensor Networks (WMSNs) [2]. Video surveillance, telemedicine, and traffic control will be the high-impact applications of emerging WMSNs.

The additional challenges created by the intrinsic features of multimedia communication must be addressed in order to deploy these multimedia applications within WSNs. Unlike conventional data communication, required for reliable transport of event features from the field, multimedia traffic does not require 100% reliability, since it is endowed with most strict requirements on bounded delay, packet loss, minimum bandwidth, and smooth change of

---

* Corresponding author. Tel.: +1 514 340 4711 x 5077; fax: +1 514 340 5139.

E-mail addresses: luis.cobo@polymtl.ca (L. Cobo), alejandro.quintero@polymtl.ca (A. Quintero), samuel.pierre@polymtl.ca (S. Pierre).

1389-1286/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2010.05.014

the transmission rate. These additional requirements inevitably amplify the challenges for multimedia communication in sensor networks. Especially, high bandwidth demands and strict multimedia communication time-constraints present significant challenges for sensor networks when matching energy and processing capacities with the level at which application objectives are met. While a significant amount of research has been conducted on WSN routing problems [3], WSN multimedia data routing remains vastly unexplored. On the other hand, multimedia communication problems have been largely investigated and numerous solutions exist for wireless environments and the Internet. However, such solutions cannot be directly applied to WMSN scenarios due to their unique characteristics and resource constraints. Consequently, there is an urgent need for research efforts to address the challenges of WSN multimedia communications to help realize many currently envisioned multimedia WSN applications.

When network size scales up, routing becomes more challenging and critical. Lately, biologically-inspired intelligent algorithms have been deployed to tackle this problem [4–8]. Using ants, bees and other social swarms as models, software agents can be created to solve complex problems, such as traffic rerouting in busy telecommunication networks. Swarm intelligence, which is revealed by such natural biological swarms, offers various valuable properties renowned in many engineering systems, for instance in network routing. Swarm intelligence systems refer to complex behaviors, typically invented from some simple agents who cooperate with one another and their environment. One of the most successful swarm intelligence techniques is called Ant Colony Optimization (ACO) [9], an optimization algorithm used to find approximate solutions for difficult combinatorial optimization problems. In ACO, artificial ants find solutions by moving on the problem graph, mimicking real ants who previously left behind pheromones for the use of future ants that can find better solutions. ACO was successfully applied to a remarkable number of optimization problems. Ants use reinforcement learning to discover the most efficient path. In reinforcement learning, the intelligent system is simply given a goal that must be reached. The system then adopts the goal using a trial and error interaction with the environment. Interactions that take the system close to the target receive a reward, while punishment is administered to those who stray away from the target. Computer scientists addressed reinforcement learning of artificial systems by introducing a concept called pheromone decay. When this pheromone evaporates rapidly, longer paths have trouble maintaining pheromone trails stable. This has also been used for telecommunication networks [10]. Artificial ants continuously explore different paths, and pheromone trails to provide backup plans. Thus, if one link breaks down, a pool of alternatives already exists.

This paper proposes a QoS routing algorithm for wireless multimedia sensor networks based on an improved ant colony algorithm. The AntSensNet protocol introduces routing modeling with four QoS metrics associated with nodes or links. The algorithm can find a route in a WMSN that satisfies the QoS requirements of an application, while simultaneously reducing the consumption of constrained resources as much as possible. Moreover, by using clustering, it can avoid congestion after quickly judging the average queue length and solve convergence problems, which are typical in ACO. Simulation results show that the proposed algorithm improves the performance of other typical protocols such as Ad hoc On-Demand Distance Vector Routing (AODV).

The remainder of the paper is organized as follows: Section 2 provides a brief review of some closely related works. The proposed protocol is described in Section 3. Then, AntSensNet is tested through a series of computer simulations presented in Section 4. Concluding remarks appear in Section 5.

2. Related work

As previously mentioned, Wireless Multimedia Sensor Networks (WMSNs) not only enhance existing sensor network applications such as tracking, home automation, and environmental monitoring, but they also enable several new applications such as multimedia surveillance sensor networks, automobile traffic management (traffic congestion avoidance systems, speed control, car parking assistance), storage of potentially relevant activities, advanced health care delivery, structural health monitoring, and industrial process control (visual inspection, automated actions). Many of these applications require the sensor network paradigm to be re-thought in view of the fact that, in these types of networks, the main concerns are not only energy constrains, limited computing power, and memory availability of the sensor nodes, but also the need for mechanisms to deliver multimedia content with a certain level of Quality of Service (QoS): the transmission of imaging and video data requires careful handling in order to ensure that end-to-end delay remains within an acceptable range, while the delay variation is acceptable, the link bandwidth pertains to the tolerable compression ratio, that jitter is satisfactory and that there is a low packet loss rate [2].

Providing QoS guarantees in wireless sensor networks consists of a very challenging problem, but several approaches have been proposed in the literature for QoS support in these kinds of networks. For example, Sequential Assignment Routing (SAR) [11], one of the first approaches in the area of QoS for wireless sensor networks, builds multiple paths from a source node to the sink node. Path selection considers both QoS metrics (the flow delay requirements and the source load balancing intentions) and energy resources, to avoid nodes with low QoS and energy reserves. Intermediate nodes forward packets according to their level of priority. However, the algorithm does not consider reliability issues and it cannot scale large networks due to the use of a routing table to calculate multiples paths.

An energy-aware QoS routing protocol for real-time traffic generated by a wireless sensor network consisting of image sensors is proposed by Akkaya and Younis [12]. This approach finds multiple network routes by using a minimum path cost. Such cost is a function of distance between nodes, node residual energy, energy transmission, and error rates which meet the requested end-to-end delay constraints. All traffic is divided into best effort and real-time classes. A Weighted Fair Queuing (WFQ) approach is used at every node to provide the required share of bandwidth for both traffic classes. Path generation is performed in a centralized manner, at the base station using an extended version of Dijkstra's algorithm. The advantage of this algorithm lies in the fact that it provides a guarantee for best-effort transmission, while simultaneously trying to maximize real-time traffic throughput. The main drawback is that the algorithm requires complete knowledge of the network topology at the base station to calculate multiple routes, thereby limiting the scalability of this approach.

The Real-time communication Architecture Protocol (RAP) proposed by Chenyang et al. [13] uses a velocity monotonic scheduler to prioritize packets, and schedules them on the basis of their required transmission speed. Geographic routing is used to forward traffic towards its destination, while velocity-aware scheduling, either static (computed at the source, based on both deadline and distance to destination) or dynamic (computed at every node, based on the actual progress of the packet) ensures that packets meet their deadlines by giving higher priority to packets with higher requested velocities. However, when calculating routes, this protocol fails to consider energy issues and the number of hops executed by the packets.

A QoS routing protocol, called SPEED, was proposed by He et al. [14] to provide a soft real-time end-to-end timeliness guarantee. The protocol requires that each node saves information regarding its neighbors and exploits geographic forwarding to find paths. In addition, SPEED strives to ensure a certain rapidity for each packet delivery so that each application can estimate the end-to-end delay for the packets by considering the distance to the sink and the speed of the packet delivery before making the admission decision. Using the distance and delay, each node evaluates the packet progress speed of each neighbor node and forwards a packet to a node whose progressive speed is higher than the pre-specified lower-bound speed. In the event that some path links become congested and cannot support the maximum delivery speed, the protocol includes mechanisms to divert traffic to other routes. One of the drawbacks of SPEED is that it does not have a packet prioritization scheme. In addition, the protocol does not provide any guarantee regarding packet reliability.

The QoS routing approach presented by Agrawal et al. [15] utilizes the geographic location of sensor nodes as well. This protocol assigns an urgency factor to every packet depending on the remaining distance and the time left to deliver the packet. It determines the distance required for the packet to be sent closer to the destination in order to meet its deadline. Each node assigns a priority to all of its neighbors, according to their residual energy and delay, as well as the priority of the packets, and packets are forwarded to the highest priority nodes. Packets are sorted in two different queues, one for non-real-time traffic, and another one for real-time traffic. Real-time traffic is prioritized based on its urgency factor, scheduling those packets with more aggressive deadlines first for transmission. Reliability is achieved by using duplication of information at the source node. However, the protocol does not consider data aggregation and the network lacks a good decongestion scheme.

A multi-path and multi-speed routing protocol called "MMSPEED" [16] is proposed by Felemban et al., which takes into account both timeliness and reliability as QoS requirements. The goal is to provide QoS support that allows packets to choose the most proper combination of service options depending on their timeliness and reliability requirements. For timeliness, multiple QoS levels are supported by providing multiple packet delivery speed guarantees. The scheme employs localized geographic forwarding with dynamic compensation to offset inaccuracies in decisions made with only local knowledge. Intermediate nodes have the ability to boost a packet's transmission speed to higher levels if they notice that the packet may not meet its delay deadline at the current speed, although the deadline could be met at a higher speed. MMSPEED assumes the use of IEEE 802.11e at the MAC layer with its inherent prioritization mechanism based on the Differentiated Inter-Frame Spacing (DIFS). Each speed value is mapped onto a MAC layer priority class. For reliability, multiple reliability requirements are supported by probabilistic multi-path routing with the number of paths being dependent upon the required degree of reliability. MMSPEED adapts to network dynamics such as channel error conditions and speed changes to determine the number of forwarding nodes (thus forming multiple paths) in each hop to satisfy the overall reliability and timeliness of QoS requirements. However, MMSPEED fails to consider energy issues; hence, it is only applicable for short-term WSN applications whose mission lasts only a few hours or at most one day. Moreover, it does not handle network layer aggregation and requires substantial state information to be stored at intermediate sensor nodes.

Finally, ReInForM [17] was proposed to address end-to-end reliability issues. ReInForM considers the importance of the data in the packet and it can adapt to channel errors. The protocol sends multiple copies of a packet along multiple paths from the source to the sink so that data can be delivered with the desired reliability. It uses the concept of dynamic packet state in the context of sensor networks to control the number of paths required for the desired reliability, based on local knowledge of the channel error rate and topology. However, the protocol only addresses QoS in terms of reliability, disregarding energy issues. In addition, the protocol does not consider route delays when selecting multiple paths.

The newly proposed solution to QoS routing in WMSNs is based on the Ant Colony Optimization (ACO) metaheuristic, especially due to the fact that the ACO algorithm [18] was inspired by the behavior of an authentic ant colony, more specifically real ants in a food search process. When ants are out searching for food, they leave their nest and walk toward the food. When an ant reaches a crossroad, it must decide which way to follow. While walking, ants deposit pheromones, leaving behind tracks of the route taken. Ants can smell pheromone and they are more likely to follow paths characterized by strong pheromone concentrations. The pheromone trails allow ants to find their way to the food source, or back to the nest. The same pheromone can be used by other ants to find the location of the food sources discovered by their mates. Based on this approach, there are many successful applications about the combinatorial optimization problems such as the Traveling Salesman Problem (TSP) [19], Vehicle Routing Problem (VRP) [20] and routing algorithms in mobile ad hoc networks [10].

According to [21], a distributed heuristic solution such as ant routing displays several features making it particularly suitable in wireless sensor networks:

- the algorithm is fully distributed; there is no single point of failure;
- the operations to be performed in each node are very simple;
- the algorithm is based on agents' asynchronous and autonomous interactions;
- it is self-organizing, thus robust and fault tolerant; there is no need to define path recovery algorithms;
- it intrinsically adapts to traffic without requiring complex, and yet inflexible metrics;
- it inherently adapts to all kinds of long-term variations in topology and traffic demand, which are difficult to take into account by deterministic approaches.

Additionally, ant routing has shown excellent performance to solve routing problems in WSNs and ad hoc networks. For instance, the Ant-Colony-Based Routing Algorithm (ARA) [22], suitable for MANETs, based both on *swarm intelligence* and ant colony meta-heuristics, consists of three phases: route discovery, route maintenance and route-failure handling. In the route-discovery phase, new routes between nodes are discovered using forward and backward ants (FAs and BAs), similar to AntNet [10]. Routes are maintained by subsequent data packets, i.e. as the data crosses the network, node pheromone values are modified so that their paths are *reinforced*. Also, same as in nature, pheromone values decay with time in the absence of such reinforcement. Routing or link failures, usually caused by node mobility, are detected through missing acknowledgments.

In [23], a mobile ant-based routing protocol for large scale WSNs is proposed. Mobile ant nodes have greater capacity in terms of communication range length, high quality multimedia sensory data processing capability, mobility management, and better energy storage. The protocol defines three types of communication patterns: sensor to ant nodes, ant to ant nodes and ant nodes to the sink. Regular sensor nodes detect the events and report to the nearest ant node(s) and the mobile ant nodes relocate them nearest to the event hotspot to capture detailed multi-modal information about the event for more accuracy. The routing protocol maintains a hierarchy of clusters and uses two types of routing tables: for intra-cluster and inter-cluster routing. The protocol is intended for upstream routing and uses only a dedicated high bandwidth backbone channel to communicate with the sink and avoid congestion. The protocol does not evaluate multimedia metrics such as bandwidth, packet loss ratio, jitter, and end-to-end delay. Also, it does not actually employ ant-based routing phenomena.

A multi-path routing protocol based on ACO intended for mobile ad hoc networks (MANET) is proposed in [24]. The protocol specializes in carrying multimedia real-time traffic over the MANET. To provide higher bandwidth and delivery guarantees, it uses a multi-path solution. It also supports high mobility for nodes and certain QoS parameters. However, the protocol uses the concept of IP-based routing and must be modified in order to be suitable for WSN.

The M-IAR protocol proposed in [25] is a flat multi-hop routing protocol that exploits the geographic location of the sensor nodes in order to select the best route possible. Basically, M-IAR finds the shortest route, the one that contains the fewest nodes between the sending and receiving nodes. The authors believe that multimedia processing is costly for resource constrained sensor nodes, in addition to the wireless communication costs. Thus, finding the shortest path with the least number of forwarding nodes will help achieve the least end-to end delay along with the best jitter conditions. However, this protocol does not differentiate packets when selecting routes. Furthermore, it ignores the concept of packet priority. Moreover, the basic assumption that shorter routes equal best routes is erroneous, especially in WMSNs with heterogeneous nodes and different link bandwidths. Finally, this protocol is unable to handle link or node failures.

Another interesting protocol is TPGF (Two-Phase geographic Greedy Forwarding) proposed in [35]. TPGF takes into account both the requirements of real-time multimedia transmission and the realistic characteristics of WMSNs. It finds one shortest (near-shortest) path per execution and can be executed repeatedly to find more on-demand shortest (near-shortest) node-disjoint routing paths. TPGF supports three features: (1) hole-bypassing, (2) the shortest path transmission, and (3) multi-path transmission, at the same time. TPGF is a pure geographic greedy forwarding routing algorithm, which does not include the face routing, e.g. *right/left hand rules*, and does not use planarization algorithms, e.g. GG or RNG. This point allows more links to be available for TPGF to explore more routing paths, and enables TPGF to be different from many existing geographic routing algorithms. But this protocol presents some inconvenience when an application wants to transmit video between a source and the sink. TPGF only takes into account to create a route the "distance" between the nodes and the sink, and other important characteristics such as link bandwidth or node queue congestion, are not considered at the moment of route discovering. Besides that, this protocol does not support heterogeneous traffic (video and scalar data at the same time).

An ant-based protocol specifically designed for WMSN is ASAR (An ant-based service-aware routing algorithm for multimedia sensor networks) presentd in [36]. This protocol define three different types of services found in a sensor networks, namely, event-driven, data query and stream query services. The ASAR chooses suitable paths to meet diverse QoS requirements from different kinds of services, thus maximizing network utilization and improving network performance. Compared to the typical routing algorithm in sensor networks and the traditional ant-based algorithm, ASAR algorithm has better convergence and provides better QoS for multiple types of services in the multimedia sensor networks. This protocol has important elements to take into account, but it lacks some others like multi-path data transmission, a very important requirement in order to increase transmission performance in WMSNs.

As mentioned above, ant-based routing algorithms exhibit a number of interesting properties for WMSN routing. However, theses algorithms have a major drawback: scalability. This problem arises since each node must send agents (*ants*) to all other network nodes in order to discover a route to the sink, meaning that the total number of agents to be sent is $N \cdot (N-1)$. In large networks, the amount of traffic generated by the ants would be excessively high. Furthermore, nodes located far from the sink increase the probability of the ants getting lost. Moreover, the ants' extensive travel times contribute to outdating the information they carry. That is the main reason why it was decided to add a clustering mechanism to the algorithm, to ensure that the protocol is more scalable and efficient. Simulation results corroborate this decision. Therefore, WMSNs will henceforth be designed based on cluster-based architecture. Nodes in the cluster are responsible for collecting scalar and multimedia data before sending this information to the Cluster Head (CH). The CH fuses such data, and then transfers the data results upstream to the sink. The sink node manages the status of CHs and broadcasts signals to the WMSNs. CHs form an independent network. They connect the sink node via multi-hop wireless links. Therefore, this paper addresses mainly the routing scheme between the CHs and sink node.

None of the existing protocols can achieve all the following goals simultaneously:

- Traffic classification, in order to differentiate network data flows, and to treat each flow with its proper QoS metrics.
- Clustering, in order to solve the scalability problems in large-scale sensor networks and to facilitate in-network processing tasks (i.e. aggregation).

3. The antsensnet protocol

AntSensNet is a specially designed routing protocol for WMSNs. It combines a hierarchical structure of the network with the principles of ACO-based (Ant Colony Optimization) routing, thus satisfying the QoS requirements requested by the applications. Besides that, our protocol supports a power efficient multi-path video packet scheduling scheme for minimum video distortion transmission.

AntSensNet comprises both reactive and proactive components:

(a) It is reactive since routes are set up when needed, not before. Once routes are set up, data packets are sent stochastically over the different paths using a pheromone table placed in each router.
(b) It is proactive due to the fact that, while a data session is in progress, paths are probed, maintained, and improved proactively using a set of special agents designed for this task.

The algorithm comprises three parts. The first constituent clusters network nodes into colonies. The second component finds network routes between clusters that meet the requirements of each application in the network using ants. The third element forwards network traffic using the routes previously discovered by the ants.

3.1. WMSNs QoS routing model

A WMSN can be presented as a connected, undirected and weighted graph $G = (V,E)$ where $V = \{v_1, v_2, \ldots, v_n\}$ denotes the set of nodes (only CHs and sink node) in the network and $E = \{e_{12}, e_{13}, \ldots, e_{xy}\}$ depicts the set of bi-directional links between CHs. For a pair of nodes v_i, $v_j \in V(i \neq j)$, if the link $e_{ij} = (v_i, v_j) \in E$ then (v_i, v_j) consists of a pair of adjacent nodes. Each node $n \in V$ in the graph, includes a set of four QoS metrics elements: $\{pl(n), ma(n), dl(n), re(n)\}$. Where $pl(n)$ expresses the maximum packet loss rate of Node n, $ma(n)$ denotes the available memory in Node n, $dl(n)$ shows the queuing delay in Node n and $re(n)$ reveals the normalized remaining energy in the node n with respect to the initial energy, defined as: $re(n) = \frac{E_{residual}(n)}{E_{initial}(n)}$, where $E_{residual}(n)$ unveils the remaining energy in the battery of Node n and $E_{initial}(n)$ indicates the initial energy in the battery of Node n. These parameters, along with bandwidth, were chosen from those mentioned by Akyildiz et al. [2], as important elements to find routes in a WMSN.

For a unicast path $P = (v_a, v_b, \ldots, S)$ from a CH v_a to the sink node s, its QoS parameters are computed as follows:

$$delay(P) = \sum_{n \in P} dl(n), \tag{1}$$

$$packetloss(P) = 1 - \prod_{n \in P} pl(n), \tag{2}$$

$$energy(P) = \min_{n \in P}\{re(n)\}, \tag{3}$$

$$memory(P) = \min_{n \in P}\{ma(n)\}. \tag{4}$$

In a WMSN, various kinds of traffic are transported by nodes. For instance, real-time audio/video data are delay-constrained with a certain bandwidth requirement. Packet losses can be tolerated to a certain extent. In addition, environmental data from scalar sensors, or non-time-critical snapshot multimedia content, are delay-tolerant and loss-tolerant kinds of data with low or moderate bandwidth demands. Finally, each type or *class* of traffic has its own requisites for QoS metrics. The goal of the AntSensNet algorithm is to find accessible paths for each *class* of traffic from a source CH node to the sink that meet different QoS requirements, thus minimizing interference among the types of traffic, balancing traffic distribution and improving network performance.

Let v_{ch} denote a CH and s denote the network sink node. The issue of routing selection from v_{ch} to s consists of finding different accessible paths P_C, where C represents each traffic class that the application being executed on the WMSN has defined. The objective function of a path P_C can be expressed as follows:

$$f(P_C) = \gamma_C^d \cdot (D_{\max} - delay(P_C)) + \gamma_C^p \cdot (1 - packetloss(P_C)) + \gamma_C^p \cdot (energy(P_C) - E_{\min}) + \gamma_C^m \cdot (memory(P_C) - M_{\min}), \quad (5)$$

where $delay(P)$, $packetloss(P)$, $energy(P)$, and $memory(P)$ respectively denote delay, packet loss rate, residual energy ratio and available memory for the path as defined in Eqs. (1)–(4). $D_{\max}$, $E_{\min}$ and $M_{\min}$ indicate the path's maximal tolerable delays, its minimal residual energy ratio and normalized memory available, respectively. Variables $\gamma_C^d, \gamma_C^p, \gamma_C^p$ and γ_C^m translate the delay weight factor, packet loss rate, residual energy ratio and available memory for global QoS parameters, respectively.

The described QoS routing problem is similar to typical Path Constrained Path Optimization (PCPO) problems, which are proved to be NP-complete [26], and an ant colony optimization based algorithm is used to solve this issue.

3.2. Assumptions

The following assumptions are made for this novel sensor network:

- Nodes are scattered randomly, in a uniform distribution, over a two-dimensional plane.
- The sink is not mobile and considered to be a powerful node endowed with enhanced communication and computation capabilities and no energy constraints.
- Sensor nodes are not mobile.
- There are two types of sensors: multimedia sensors (resource-rich nodes, capable of audio/video sensing of their environment) and scalar sensors that capture data such as temperature, light or pressure. Both types are also distributed randomly over the area.
- Nodes are unaware of their location, i.e. they are not equipped with a GPS device.
- Communication from each node follows an isotropic propagation model.
- Radio transmitting power is controllable, i.e. nodes can adjust the transmitting power according to the distance.
- Despite the fact that nodes are heterogeneous, it is assumed that radio transmissions are identical for all nodes.
- Nodes can estimate the approximate distance by the received signal strength, given the transmit power level is known, and the communication between nodes is not subject to multi-path fading.
- We use the same radio model presented in [27], and it is assumed that the radio channel is symmetric so that the energy required to transmit a *m*-bit message from Node *i* to Node *j* is identical to the energy required to transmit a *m*-bit message from Node *j* to Node *i*, for a given signal to noise ratio.

3.3. Clustering process

AntSensNet is a QoS routing protocol based on an ant colony algorithm. This algorithm makes use of special agents (known as *forward-ants* or FANTs) to find a path between a sensor node and the network base station or *sink*. In the route discovery process, several ants leave their node source, aiming for their neighbors, each one with the task of finding a route, meaning that sensor nodes must communicate with one another and the routing table of each node must contain the identification of all sensor nodes in the neighborhood as well as their corresponding levels of pheromone left on the trail. As the number of nodes grows, the number of agents required to establish the routing infrastructure may explode [28]. A way to overcome the overhead explosion and reach scalability consists of using the hierarchical routing approach.

Nevertheless, scalability is not the only reason to cluster the network. This process also allows for improving network data aggregation mechanisms, while concentrating this activity in the CH, consequently reducing node workloads, saving energy and increasing the network lifetime. Arboleda and Nasser [29] presents other advantages of clustering that apply to this novel protocol: the fact that only the CH transmits information out of the cluster helps prevent collisions between the sensors inside the cluster, as they do not have to share communication channels with nodes in other clusters. This also promotes energy savings and avoids the black hole problem. Latency is also reduced. Although data must hop from one CH to another, they cover larger distances than when sensors use a multi-hop communication model (non-clustered) as the one used in other protocols.

Finally, clustering is applied in order to take advantage of the existence of nodes of different abilities inside a WMSN. Table 1 [30] presents the processing performance and memory capacities among standard (TelosB) and multimedia sensors. Table 1 shows that the memory and processing capacities of multimedia sensors are superior to those of conventional sensors. That is the reason for selecting multimedia sensors to become the network CHs. This novel algorithm will be designed to favor the "selection" of these nodes as CHs.

Our clustering algorithm aims at achieving the following goals:

- Saving network resources by encouraging the selection of resource-rich nodes (multimedia sensor nodes) as network CHs.
- Ensuring network connectivity by forming a virtual backbone among the different CHs. Each CH is in the radio range transmission of at least one other CH. Communication between two CHs is direct (there are no relay nodes between them).
- Maximizing network lifetime by implementing a mechanism of CH rotation.

With a virtual backbone in the network, only CHs are concerned with data transportation, and other nodes are free to pursue their sensing tasks. Such task sharing improves network performance with respect to routing overhead and, moreover, a smaller number of nodes need to be alert for data transportation. This procedure reduces energy consumption, thus simultaneously maximizing network lifetime.

Table 1
Abilities of video and standard sensors.

	Stargate	Samsung S3C44B0X	TelosB
Clock frequency	200/300/400 MHz	66 MHz	8 MHz
Architecture	32 bit RISC	16/32 bit RISC	16 bit RISC
Memory	64 MB SDRAM 32 MB Flash	256 MB	10 KB 1 MB Flash
Cache	32 KB data 32 KB instruction	8 KB	Data not Available
Cost ($)	595	500	100

3.3.1. Information update phase

This novel clustering algorithm is based on T-ANT [28] and the clustering protocol uses a collection of agents to form clusters in a sensor network. It is completely distributed and completed in constant time. These are the reasons why this algorithm was selected.

As in T-ANT, clustering operations are split into rounds. Each round comprises a cluster setup phase and a steady phase. In the steady phase of the algorithm, data transmission takes place between sensors and the sink. A number of timers are used to control the process operations. During the cluster setup phase, CHs are elected and clusters are placed around them. In order to avoid the maintenance of many state variables, as one finds in numerous current clustering proposals, a series of agents (known as *cluster-ants* or CANTS) are used to control CH elections. A node with a CANT becomes a CH, whereas others choose to join the best cluster in range.

The cluster radius $R_{cluster}$ is defined as a tunable parameter that determines the minimum distance between any two CH nodes in the network. The value of this parameter always remains inferior to the sensor communication radio range (called r). Before the cluster setup phase, an information update phase is carried out by the sensors. Each sensor node broadcasts a `HELLO` packet with information regarding its ID, its *clustering pheromone value* ($\Phi_c(n)$) and its *state* to its neighbors. When a `HELLO` packet arrives, the node stores such information in a table, the *neighborhood* or the neighbor's information table. This table is then used to select clusters, to join a cluster and to route data packets.

The clustering pheromone value determines whether it is appropriate for this node to become a CH. For each node, this value is calculated using the following formula:

$$\Phi_c(n) = (ma(n)t)^a \cdot (re(n))^b, \qquad (6)$$

where $ma(n)$ denotes the available memory in the node, $re(n)$ is the residual ratio of the node's energy and a and b denote the importance of each component of the pheromone: a for the memory capacity and b for the energy component. Thus, the application determines which component is most important when selecting a CH, namely, memory or energy or both.

The state indicates if the node is a CH or a member of a cluster or neither. These `HELLO` packets are constantly broadcast by the nodes throughout their lifetime.

3.3.2. Ant release phase

After the information update phase, the sink releases a fixed number of ants (i.e. control messages) into the network. Assuming that the terrain is square, $M \times M$, the number of ants to be released is set at $\left\lceil \frac{M^2}{\pi d^2} \right\rceil$, where d depicts half of $R_{cluster}$. The latter formula also represents the number of clusters that make up the network. Attempts are made to obtain complete coverage of the area with this number of clusters, where every node belongs to a cluster and the CHs are disseminated throughout the terrain. Ants move about the network in a random fashion, as far as they can, respecting the limits imposed by their Time-To-Live (TTL) values. The TTL value equals the number of ants. Hence, an ant can visit a large number of candidate nodes to become a CH before they die. When the sink releases an ant, it chooses one of its neighbors randomly according to the following probability distribution function:

$$prob_c(j) = \frac{\Phi_c(j)}{\sum_{i \in N_s} \Phi_c(i)}, \qquad (7)$$

where $\Phi_c(j)$ denotes the clustering pheromone value sent by Node j, as defined in Eq. (6), and N_s represents the set of all of the sink neighbors located at a distance of at least $R_{cluster}$. Before releasing the next cluster ant, the sink waits for a timer to expire (`CLUSTER_TIMER`). Although the timer expiration is set at a random value, it always remains proportional to the delay of sending an ant from a node to a neighbor. The objective of this timer is to ensure that the ants' subsequent transmissions do not self-interfere. Aside from that, when the sink selects a neighbor, the pheromone value of that node is artificially decreased, in order to avoid choosing the same set of nodes repeatedly.

Algorithm 1 presents the tasks performed by the sink in order to start the clustering process.

Algorithm 1. Tasks developed by the sink

```
$d \leftarrow \frac{R_{cluster}}{2}$
repeat
    Use probability distribution function ($prob_c$) to choose a neighbor ($i$)
    Send a cluster ant to node $i$ with a $TTL = \left\lceil \frac{M^2}{\pi d^2} \right\rceil$
    Wait until a CLUSTER_TIMER expires
until all ants are released
```

When an ant arrives at a node, that node will execute the tasks depicted in Algorithm 2:

Algorithm 2. Tasks developed by the other nodes

```
if an ant arrives at node $i$ then
  if node $i$ is not a CH then
    if there is a CH in the radius $R_{cluster}$ then
      Pick a random CH neighbor
      Send the ant to it
    else
      Store the ant {This node is a CH}
      Broadcast a message ADV_CLUSTER to neighbors in range $R_{cluster}$
    end if
  else if node $i$ is a CH then
    Decrement the TTL of the ant
    if $TTL > 0$
      Pick a random neighbor according to the probability function $prob_c$
      Send the cluster ant to it
    else
      Destroy the ant
    end if
  end if
end if
```

Algorithm 2 shows that, in order to become a CH, the selected node must have received a *cluster ant* from another CH (or the sink) located at a distance $R_{cluster}$ from it. $R_{cluster}$ was previously defined as the minimal distance between two CHs. Hence, at the moment of selecting the following neighbor, the node reads its neighbors' information table and selects, with probability $prob_c$, a node whose distance is a minimum of $R_{cluster}$ in a random manner. The reason why a CH elects the next CH is to create a virtual backbone between the various CHs, a direct communication strategy between them. This backbone will facilitate the task of routing accomplished by the protocol AntSensNet. When a node becomes a CH, it broadcasts an `ADV_CLUSTER` message to advise its neighborhood of its new condition. It also changes the value of field *state* of a `HELLO` packet subsequently sent by the node. Once a regular node receives an `ADV_CLUSTER` message from a CH located at a distance below $R_{cluster}$, it stores the corresponding information that pertains to that CH. This information is later used to join a given cluster. Contrary to other proposals documented in the literature, this CH election approach has a very small constant time and a low level of complexity.

An ant's TTL indicates the maximum number of hops that it can perform. The CH pulverizes an ant once its TTL reaches the value zero. This situation shows the existence of a superfluous number of clusters in the network, and the cluster ant is destroyed in order to avoid the creation of new clusters that would hinder the network.

The actual clustering process happens once another timer expires. A regular node decides to join a cluster when its `JOIN_TIMER` expires. This node chooses the nearest cluster to join (from all of the `ADV_CLUSTER` packages it received) by sending a `JOIN` message with its ID. When a CH receives `JOIN` messages, it stores such information in order to subsequently select a cluster member as a new CH. If a regular node has never received an `ADV_CLUSTER` package from a CH, it starts a `JOIN_TIMER` once again and repeats the latter process until this timer expires. However, if in the process, it receives neither an `ADV_CLUSTER` message nor a `HELLO` package from a CH, the node uses the nearest neighboring cluster member as a "bridge" to reach its CH.

When a CH realizes that the node μ is three or more hops away from it, that CH selects the neighbor in the path to μ as a new CH. This new CH broadcasts an `ADV_CLUSTER` message in order to contact other CHs and initialize their pheromone tables. In that way, we can get a better CH distribution to cover the whole network area. This new CH selection may be done in any moment in the protocol execution.

The properties of the proposed clustering algorithm can be highlighted as follows:

(1) The algorithm is completely distributed. A node locally decides to become a CH if an ant reaches it or joins a cluster.
(2) Given the absence of looping statements as a function of node quantity, it is clear that the election process has an $O(1)$ time complexity.
(3) The algorithm ensures the creation of a backbone among the CHs. As all CHs are connected, paths to a sink can be easily discovered.

Fig. 1 shows an example of a sensor network clustering using our algorithm.

3.4. AntSensNet algorithm description

AntSensNet consists of a protocol based on ACO (Ant Colony Optimization) to discover and maintain routes between CHs and the sink. The route discovery process starts as soon as the cluster process finishes. Before presenting the algorithm, here are some definitions.

3.4.1. The ant's structure

The data configuration of the ant's structure used in its route discovery process is defined below. It comprises the following fields:

(a) *ant.ID*: the ant's ID.
(b) *ant.type*: the type of ant in the route discovery process. This field can be a FANT (*forward ant*), a BANT (*backward ant*), a MANT (*maintenance ant*) or a DANT (*data ant*).
(c) *ant.nodes*: the nodes-visited-stack, contains the IDs of nodes by which the ant passes.
(d) *ant.hopcount*: calculates the number of hops by which the ant passed from its CH source. This field serves as the ant's TTL.
(e) *ant.info*: Each type of ant uses this field to store special information about the route or the nodes, in order to evaluate how appropriate the route is. This field contains the following subfields:

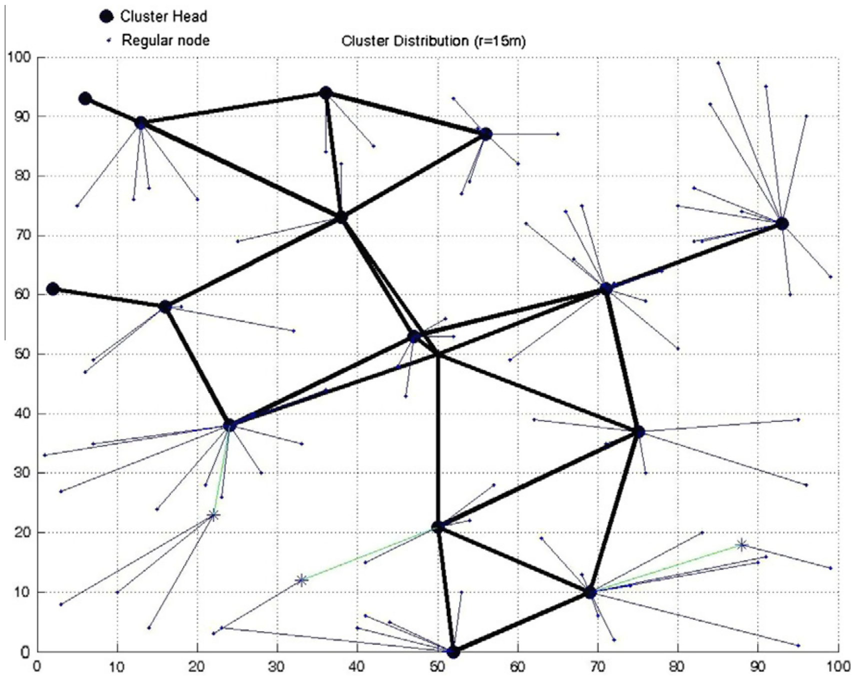


Fig. 1. WMSN clustering example. The backbone created by CHs is outlined in black.

- The minimum residual energy of the nodes by which the ant passed.
- The cumulative queue delay, packet loss, and available memory of each node visited by the ant.

3.4.2. The queuing model

Sensor data may originate from various types of sources whose levels of importance vary. Akyildiz et al. [2] organizes the following examples of traffic into various WMSN classes:

- *Real-time, loss-tolerant, multimedia streams*. This class includes video, audio or multi-level streams composed of video/audio and other scalar data (e.g. temperature readings), as well as metadata associated with the stream that need to reach a human or an automated operator in real-time, i.e. within strict time limits, although relatively loss tolerant (e.g. video streams can tolerate a certain level of distortion). Traffic that belongs to this class is usually associated with high bandwidth demands.
- *Delay-tolerant, loss-tolerant, multimedia streams.* This class includes multimedia streams intended for storage or subsequent offline processing, whose delivery is not bound by strict delays. However, due to the typically high bandwidth demands of multimedia streams and due to limited buffers of multimedia sensors, data that belong to this category need to be transmitted virtually in real-time in order to avoid excessive losses.
- *Delay-tolerant, loss-intolerant, data.* This may include data from critical monitoring processes, with low or moderate bandwidth demands that require some form of offline post processing.
- *Delay-tolerant, loss-tolerant, data.* This may include environmental data from scalar sensor networks, or non-time-critical snapshot multimedia content, with low or moderate bandwidth demand.

Hence, packet scheduling policy should consider different priorities (importance) for different types of traffic classes. Fig. 2 shows the queuing model for a sensor considering different traffic classes. At the outset, the application must define these classes and their parameters, i.e. minimal energy, bandwidth, available memory and packet delays, and maximum packet loss. The application, rather than the protocol, is responsible for predefining the number of classes. The application is also responsible for assigning the class and priority of every packet sent by the sensors. For each CH, a classifier checks the class of the incoming packets which are then sent to the appropriate queues, and a scheduler organizes packets according to their classes and level of priority.

The fact that an application can define the classes of data for the queue enables a great flexibility but it should be used with caution. This characteristic would create problems when the application defines too many classes of traffic, it will degrade the performance of the nodes because of big utilization of memory.

3.4.3. Pheromone table

An ant pheromone table is a data structure that stores pheromone trail information for routing from Node *i* to the sink via a CH neighbor *j*. Saved into the node memory, this structure is organized as shown in Table 2.

In Table 2, each column reflects the different traffic class, as defined by the application. Each row corresponds

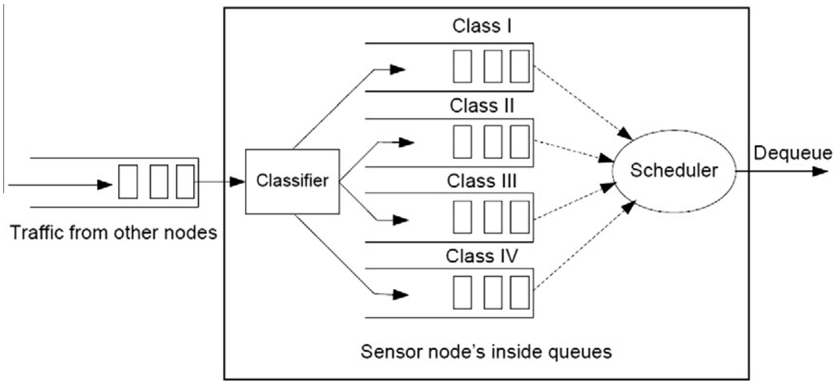


Fig. 2. Queuing model on a multimedia sensor node.

Table 2
Pheromone table for Node *i*.

Neighbor	Traffic Class k				Class t	Expiration time
N_1	$e_i^k(1)$	$\delta_i^k(1)$	$\varepsilon_i^k(1)$	$\mu_i^k(1)$	$\cdots$	T_1
N_2	$e_i^k(2)$	$\delta_i^k(2)$	$\varepsilon_i^k(2)$	$\mu_i^k(2)$	$\cdots$	T_2
$\vdots$	$\cdots$				$\vdots$	$\vdots$
N_j	$\cdots$				$\vdots$	$\vdots$

to a neighbor. There are four values for each traffic class in the table. Each value is a pheromone trail concentration for each QoS metric used by the protocol:

(1) $e_i^k(j)$: energy pheromone value from link between Nodes i and j for packets that belong to traffic class k;
(2) $\delta_i^k(j)$: delay pheromone value from link between Nodes i and j for packets that belong to traffic class k;
(3) $\varepsilon_i^k(j)$: packet loss pheromone value;
(4) $\mu_i^k(j)$: available memory pheromone value.

Every entry in the pheromone table has an expiration time and certain entries are disabled as time goes on. When the current time exceeds the set expired time, a new route-discovery phase commences.

3.4.4. Route discovery

When a regular node needs to send data to the sink, such information is immediately sent to its CH. The working process of the AntSensNet algorithm is described as follows: when a CH node is in possession of sensor data to be sent, it checks its routing table to find an appropriate path for the traffic class of the packet. Before initiating a data transmission, the CH source checks out its pheromone table in order to find any non-expired node information. That information is expired if the value associated to the *Expiration Time* field is inferior to the node clock. If all the information in the pheromone table is expired, a new route probe phase is started. There are a number of forward ants needed to send for route probes. After the routing discovery process, cached data are immediately sent to their destination. To reduce delays associated with the first discovery phase, an AntSensNet algorithm launches a full route probe phase for each traffic class after the clustering process was ended. A packet flow that shows a CH receiving an ant is illustrated in Fig. 3. There are three phases to the AntSensNet: the forward ant phase, the backward ant phase and the route maintenance phase.

Forward ants phase: If a CH finds that there is no satisfactory and unexpired path to the sink in the packet's traffic class in its routing table, it generates a certain number of Forward Ants (FANTS) to search for paths leading to the sink. Forward ants are agents that establish the pheromone track from the source CH to the sink node. The ants' structure is presented above. In their info field FANTs carry:

- The minimum residual energy (*energy*) of the nodes by which the ant passed;
- The cumulative queue delay (*delay*), packet loss (*packetloss*) and available memory (*memory*) of each node the ant visited.

These values are the QoS metrics used in order to discover routes. To find a route to the sink, the CH source broadcasts a FANT. Each field of the ant packet must be set before being sent, i.e. the type field $ant.type \leftarrow FANT$, $ant.hopcount \leftarrow 0$, and it pushes the CH source into the *ant.nodes* stack. When an intermediate CH receives a FANT, it judges the existence of loops on the *ant.nodes* field of the received FANT. Those ants resulting in route loops are discarded. Before sending the FANT to the next CH, the field *ant.info* must be updated with local information regarding the current CH. This update is carried out in the following way:

$$energy \leftarrow \min(energy, re(CH))$$
$$delay \leftarrow delay + dl(CH)$$
$$packetloss \leftarrow packetloss \times pl(CH)$$
$$memory \leftarrow \min(memory, ma(CH))$$

When a CH receives a FANT, it updates the info field of the ant, increments the ant's hop count and push its identification (e.g. i) on the ant's node stack. The next hop is selected according a certain probability value. The

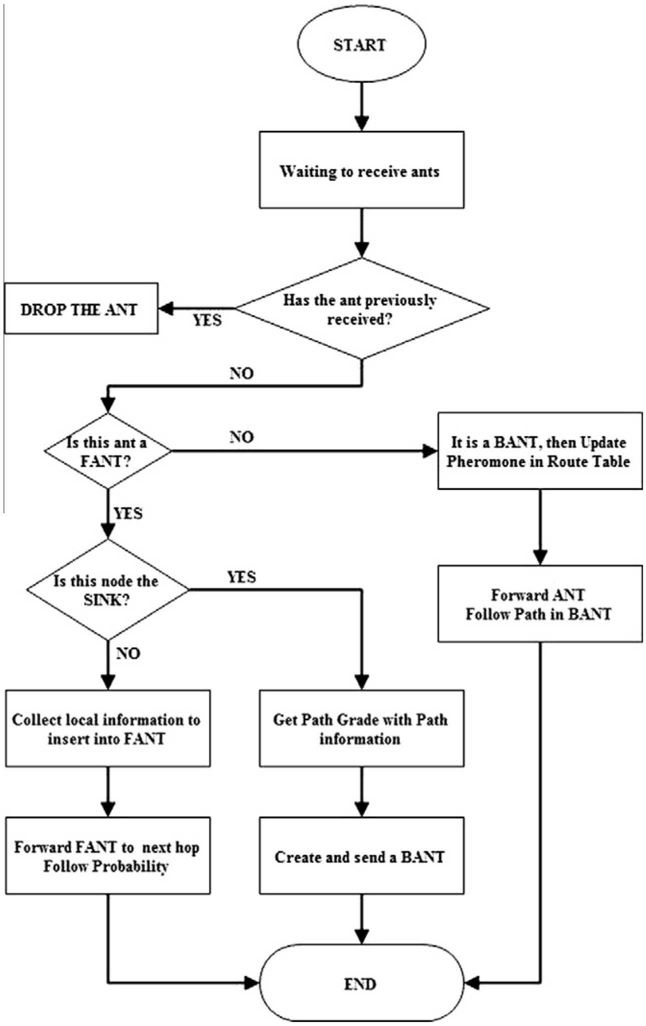


Fig. 3. Route discovery process of AntSensNet.

probabilistic value $P_i^k(j)$ determines the probability of moving from CH i to j for the traffic class k, which is computed as expressed by Eq. (8):

$$P_i^k(j) = \begin{cases} \frac{\Psi_i^k(j)}{\sum_{s \notin V_{pass}} \Psi_i^k(s)} & \text{if } j \notin V_{pass}, \\ 0 & \text{if } j \in V_{pass}, \end{cases} \tag{8}$$

where V_{pass} is the set of nodes that the FANT has passed. $\Psi_i^k(j)$ is the normalized value of pheromone from i to j for the traffic class k. This value combines all of the QoS parameters the application has established for the traffic class. In order to compute this value, the following probability value must be calculated:

(1) The normalized energy probability:

$$p_{e,i}^k(j) = \frac{e_i^k(j)}{\sum_{s \in N_i} e_i^k(s)},$$

where N_i indicates the set of CH neighbors of CH i and $e_i^k(j)$ is the energy value for Node j and traffic class k in the pheromone table of Node i.

(2) The normalized delay probability:

$$p_{\delta,i}^k(j) = \frac{\delta_i^k(j)}{\sum_{s \in N_i} \delta_i^k(s)},$$

where N_i denotes the set of CH neighbors of CH i and $\delta_i^k(j)$ depicts the delay value for Node j and traffic class k in the pheromone table of Node i.

(3) The normalized packet loss probability:

$$p_{\varepsilon,i}^k(j) = \frac{\varepsilon_i^k(j)}{\sum_{s \in N_i} \varepsilon_i^k(s)},$$

where N_i identifies the set of CH neighbors of CH i and $\varepsilon_i^k(j)$ is the packet loss value for Node j and traffic class k in the pheromone table of Node i.

(4) The normalized available memory probability:

$$p_{\mu,i}^k(j) = \frac{\mu_i^k(j)}{\sum_{s \in N_i} \mu_i^k(s)},$$

where N_i translates the set of CH neighbors of CH i and $\mu_i^k(j)$ is the memory value for Node j and traffic class k in the pheromone table of Node i.

Finally, the normalized pheromone value from i to j for the traffic class k, $\Psi_i^k(j)$, is calculated as:

$$\Psi_i^k(j) = \frac{\alpha_e p_{e,i}^k(j) + \alpha_\delta p_{\delta,i}^k(j) + \alpha_\varepsilon p_{\varepsilon,i}^k(j) + \alpha_\mu p_{\mu,i}^k(j)}{\sum_{s\in N_i}\left[\alpha_e p_{e,i}^k(s) + \alpha_\delta p_{\delta,i}^k(s) + \alpha_\varepsilon p_{\varepsilon,i}^k(s) + \alpha_\mu p_{\mu,i}^k(s)\right]}. \tag{9}$$

Note that Ψ is calculated as the addition of all QoS parameters collected by the ants, that is, the energy, delay, bandwidth, packet loss and available memory pheromones, normalized into a single quantity with a comparable magnitude. Normalizing pheromones makes it possible to convert them into the same dimension. Note that *alpha* values are arbitrary, positive constants, which represent the importance of each QoS components in the selection of the next hop in the route.

3.4.4.1. Backward ants phase. When a forward ant reaches the sink, the evaluation of the found route is carried out. The information collected by the FANT is compared with the parameter values set by the application for each QoS metric. For instance, the application can demand routes with a packet loss value that is inferior to 1% and a residual energy ratio superior to 80%. The sink evaluates the FANT's info versus these parameters and determines whether the route is adequate. If the route does not fulfill the application requirements, the FANT is discarded. The application must tune these parameters in order to obtain efficient routes. The sink may reject all of the paths found by the ants if parameters are unreal or impossible to obtain under the current network conditions.

When an appropriate FANT is received that meets the application requirements, the sink pulverizes the FANT and a backward ant (BANT) is generated. A BANT carries the collected information of its corresponding FANT and the path's intermediate node IDs and it is sent back using the reverse path of its corresponding FANT. When a BANT is received at intermediate CH i, the information stored inside such BANT is used to update the pheromone value and hence the probability routing table entry corresponds to the FANT's destination. The pheromone values on the incoming link are increased and the values pertaining to the other links are decreased using the pheromone update functions. These functions work as follows:

(1) For the energy pheromone:

$$e_i^k(j) = \begin{cases} \rho_e \cdot energy + (1-\rho_e)\cdot e_i^k(j) & \text{incoming link,} \\ (1-\rho_e)\cdot e_i^k(j) & \text{other links,} \end{cases} \tag{10}$$

where $e_i^k(j)$ depicts the pheromone value corresponding to residual energy for the traffic class k and Neighbor j at Node i, *energy* is the collected value by the corresponding FANT about the minimal path's residual energy and $\rho_e(0 < \rho_e < 1)$ is the pheromone improvement parameter for the incoming link. Its purpose is to enforce efficient routes while decreasing the appropriateness of the bad ones (pheromone evaporation). Other pheromone update functions are similar.

(2) For the delay pheromone:

$$\delta_i^k(j) = \begin{cases} \frac{\rho_\delta}{delay} + (1-\rho_\delta)\cdot \delta_i^k(j) & \text{incoming link,} \\ (1-\rho_\delta)\cdot \delta_i^k(j) & \text{other links,} \end{cases}$$

where $\delta_i^k(j)$ denotes the delay pheromone value stored in the CH i for Class k and Neighbor j, and *delay* represents the delay value collected by the corresponding FANT. Likewise, in the energy pheromone formula, $\rho_\delta(0 < \rho_\delta < 1)$ represents the pheromone improvement factor for the incoming link of the BANT and $(1-\rho_\delta)$ represents the pheromone evaporation factor for the others links.

(3) For the packet loss pheromone:

$$\varepsilon_i^k(j) = \begin{cases} \frac{\rho_\varepsilon}{packetloss} + (1-\rho_\varepsilon)\cdot \varepsilon_i^k(j) & \text{incoming link,} \\ (1-\rho_\varepsilon)\cdot \varepsilon_i^k(j) & \text{other links,} \end{cases} \tag{11}$$

where $\varepsilon_i^k(j)$ shows the packet loss pheromone value stored in the CH i for Class k and Neighbor j, and *packetloss* represents the packet loss value collected by the corresponding FANT. Similar to the delay pheromone formula, ρ_ε $(0 < \rho_\varepsilon < 1)$ represents the pheromone improvement factor for the incoming link of the BANT and $(1-\rho_\varepsilon)$ represents the pheromone evaporation factor for the others links.

(4) Finally, for the available memory pheromone:

$$\mu_i^k(j) = \begin{cases} \rho_\mu \cdot memory + (1-\rho_\mu)\cdot \mu_i^k(j) & \text{incoming link,} \\ (1-\rho_\mu)\cdot \mu_i^k(j) & \text{other links,} \end{cases}$$

where $\mu_i^k(j)$ indicates the memory pheromone value stored in the CH i for Class k and Neighbor j, and *memory* represents the packet loss value collected by the corresponding FANT. Similar to the delay pheromone formula, $\rho_\mu(0 < \rho_\mu < 1)$ represents the pheromone improvement factor for the incoming link of the BANT and $(1-\rho_\mu)$ expresses the pheromone evaporation factor for the others links.

The pheromone trails of the best route offer incentives, by providing a greater amount of pheromone. Furthermore, in the current of history, the worst route offers a pheromone punishment incite other ants to stay away from the worst solution.

The BANT is sent to the next CH in the reverse path of the corresponding FANT. When the BANT reaches the FANT's source CH, it is pulverized after updating the pheromone table of the source table. Data can then be sent to the sink following the maximum probability path.

Routing maintenance phase: While a node sends information that belongs to a given traffic class, FANTs are generated periodically in order to find updated routes, i.e. topology changes in the network that fulfill the QoS requirements specified by the application. The process of routing maintenance also deals with congestion and lost link problems.

(1) *The Congestion Problem:* When the load of a queue of a traffic class at an intermediate CH surpasses a predefined threshold (called Γ), the CH sends a congestion-control MANT to its upstream neighbor nodes

to modify the pheromone tables for the given traffic class. The TTL of this MANT is set according to the severity level of the traffic: the heavier the traffic, the higher the TTL value. Upon receiving the MANT from the CH *j*, Node *i* reduces the strength of pheromone on the corresponding route and traffic class. Then, Node *i* uses other routes with a relatively high level of pheromone to forward packets that are part of the congested traffic class.

(2) *The Lost Link Problem:* AntSensNet also uses periodic `HELLO` messages to update information about the connectivity of neighboring nodes. Once the next hop becomes unreachable, the CH first deletes all the entries, in the pheromone table of Node *i*, which correspond to the broken link, and then searches its pheromone table for an alternative neighbor node for subsequent data transmissions. The CH then sends a MANT to all neighbors in order to inform them that Node *i* is unreachable and that it must be removed from their pheromone tables.

Data transmission phase: In AntSensNet, a CH forwards data following the maximum pheromone value path. When a node has multiple next hops for a given traffic Class *k*, it selects one with the maximum Ψ. This value is calculated in the same way as that of a FANT, Eq. (9). This strategy leads to data loading spreads according to the estimated path quality. When estimates are kept up-to-date, which is done by using the FANT, as described in the previous section, *automatic load balancing* ensues. When a path is clearly worse than another, it is avoided, thus reducing its traffic load. Other paths thus obtain more traffic, causing greater congestion, thus reducing their QoS parameters. Continuously adapting data traffic incites nodes to spread data loads evenly over the network.

Data Ants or DANTs: In AntSensNet, ants are special agents that assist in route discovery and maintenance. However, they are also high priority packets. They are sent, processed, and received by the CH with a higher priority than any other traffic class. A special ant, known as DANT (Data Ant), is assigned to transport urgent (or real-time) data from a node to the sink. In this case, the information is encapsulated in this special type of ant, and it is processed before all of the other traffic classes in every node. The behavior of DANTs is similar to that of FANTs, yet the former do not collect information from each CH they meet along their route, nor do they generate BANTs when they arrive at the sink. Also, they choose their next hop according to the path that has the maximum level of pheromones.

3.4.5. Video transmission

If an application needs more accuracy to transmit video, AntSensNet offers a mechanism to transport a video stream between a source node and the sink. This mechanism use an efficient multi-path video packet scheduling scheme for minimum video distortion over the wireless network. The mechanism is based on the “Baseline” algorithm proposed in [34]. That scheme uses H.264/AVC codec as encoding technique because of its compression efficiency, low complexity and error resiliency.

In that paper, the authors express that the high end-to-end bandwidth requirements of video communication usually cannot be met by the WMSNs, when the traditional single-path routing approach is used, leading to perceived video quality degradation. In order to meet the QoS requirements, a multi-path approach should be adopted, where the video source delivers the data to its destinations via multiple paths, thereby supporting an aggregated transfer rate higher than what is possible with any one path. Specifically, the encoded video data are segmented and multiplexed in a specific way, based on their distortion importance, over different paths so that the sink can assemble the video data and decode them with the maximum perceived quality. AntSensNet, on an application demand, is able to create multiple paths to transport video packages. In other words, the protocol has the possibility to send the video packets using a single-path or a multi-path scheme, based on an application decision.

Multiple paths to the sink: Multi-path video transmission has been studied extensively [37]. The benefits of selecting multiple paths among a video server and a client instead of just the shortest path include among others:

- reduced correlation among packet losses;
- increased channel resources that can support the application’s demands in QoS;
- the power consumption is more evenly spread in the network nodes preventing node failures;
- ability to adjust to arbitrary congestion occurrences in different parts of the network.

When a video source wants to initiate a video transmission and its CH does not have an active route to the sink, that CH source initiates route discovery by broadcasting a special video forward ant (VFANT) packet to the CH neighbors. The behavior in each intermediate node is the same as when discovering a single-path. Unlike to the single-path routing algorithm, in order to discover multiple paths, intermediate nodes do not discard duplicate VFANTs. When a VFANT reaches the sink, it generates a video backward ant (VBANT) packet for the CH source node. The VBANT returns to the source using the nodes that corresponding VFANT visited. The routing table update methods are the same as single-path discovery. Since duplicate VFANTs are not discarded, the sink node may send multiple VBANTs back to the source. At the source, received VBANTs are examined and those that do not provide link-disjointness with the routes discovered by other VBANTs are discarded. After that, the CH source has a set of link-disjoint paths to use in a video transmission. When the video packet sending starts, the next hop is determined by the discovered routes and these routes are only modified when problems like congestion, like failures, etc., emerge.

Video distortion model: We use the Baseline schedule algorithm proposed in [34]. This algorithm firstly identifies the possible paths from the CH sender to the sink that can on aggregate satisfy the quality of services requirements of the video service. Secondly, in case that the aggregate bandwidth of the multiple paths is limited, the algorithm utilizes the following video distortion prediction model to determine the least important packets that could be dropped prior to transmission.

In order to analytically express the distortion model, a list of previously encoded reference frames with size M_{REF} that is used during the encoding and decoding processes for motion-compensated prediction, is defined. This parameter accounts for the impact of the number of reference frames on the distortion propagation. Moreover, each frame is coded into a number of video packets according to each size. Finally, a simple error concealment mechanism, which replaces a lost frame with its previous at the decoder, is applied. The proposed model includes analytical models for a single frame loss, a burst of losses with variable burst length B (where $B \geqslant 2$) and frame losses separated by a lag.

Baseline packet scheduling: The authors of [34] introduce the "Baseline" packet scheduling algorithm. In our implementation, this algorithm use AntSensNet in order to transport the packages between a CH and the sink. Under these conditions the "Baseline" packet scheduling algorithm schedules the transmission of video packets via multi-paths by dropping the excess video traffic in order to prevent network congestion.

In more details, channel resources in a WMSN are scarce and there are cases when the transmission requirements exceed the available aggregate transfer rate of the multiple paths. If the required rate for error free transmission (RTR) is higher than the current available aggregate transmission rate (ATR) then the sender decides which video packets will be optimally dropped in order to adapt its current rate to the allocated one. The packets to be dropped are selected according to their impact to the overall video distortion. A combination of one or more video packets may be omitted prior to the video transmission by the video source. Dropping a packet imposes a distortion that affects not only the current video frame but all the correlated video frames. The intelligence of the packet scheduling algorithm is that utilizes the distortion prediction model presented previously, which considers the correlation among the reference frames, thus it selects the optimum pattern of packets to drop in each transmission window.

This process is neither time nor power consuming, as the transmission window is generally small and the mathematical calculations are not of high complexity. The transmitted packets are distributed among the available routes according to their impact in the video distortion; hence packets of high importance are transmitted through the higher capacity routes.

4. Experimental results

Three main aspects of AntSensNet were evaluated: its clustering process, its routing algorithm, and the video transmission mechanism, which were analyzed separately. In each instance, an NS-2 [31] was used to implement and simulate the novel algorithms. There are two types of nodes: scalars and multimedia (with more energy and memory than scalar nodes). Half of the nodes are multimedia. The radio range of the nodes spans 100 m and the data rate equals 2 Mbit/s. At the MAC layer, a modified version of 802.11b DCF protocol was used. The modification was made in the queue politics of the MAC protocol in order to accept multi-class and multi-priority traffic.

4.1. The clustering process

For these simulations, it was assumed that 100 sensor nodes were distributed randomly over a square area of 100 m × 100 m. This scenario was executed during 600 s. In order to benchmark this new protocol, it was decided to compare it to T-ANT [28], as it was the base of our clustering protocol and also since it outperformed other well-known clustering algorithms, such as LEACH [27] and HEED [32]. For this experiment, an $R_{cluster} = 20$ m is assumed, and the same CH rotation scheme is used in AntSensNet as in T-ANT: there are multiple rounds in the network lifetime, and in each round, a CH rotation is carried out. The CH finds its cluster member whose level of pheromone, Eq. (6), is the highest, before it becomes the CH for the next round.

Fig. 4 depicts the CH connectivity of these protocols at different simulation time. This property indicates if there is direct communication between the CHs of the network, meaning that no CH isolated. This property is very important in this novel algorithm, as all of the traffic between source nodes and the sink is transported by the CH. If CHs are isolated, it is impossible to transmit information from that cluster to the sink. In this simulation, any node can be a CH, in other words we set $a = b = 0$ in Eq. (6). Observe that after only 20 rounds (one round 20 s each), the connectivity of T-ANT is acceptable. Meanwhile, the connectivity of AntSensNet remains at a steady 100%. The main design goal of our clustering algorithm is reached with the permanent connectivity of the CHs.

In Fig. 5, the improvement gained through our AntSensNet clustering algorithm is further exemplified by the network lifetime graph. The network lifetime is defined as the time the first node in the network has a depleted battery. For this experiment, the memory component in the clustering pheromone formula (parameter a in Eq. (6)) was set to *0* (zero) and the energy component (parameter b in Eq. (6)) was set to *1*. This way, energy rich sensors have greater probabilities of becoming a CH. Moreover, a Constant Bit Rate CBR traffic source was used to generate data traffic of 32-byte packets. All regular nodes sent the sink a packet/s on average. Non sending nodes fall into a sleep mode. Five simulations were carried out, where the value of the energy component varied. The initial energy of the scalar nodes was 0.1 J and for the multimedia nodes, this initial energy was 0.5 J in order to let the nodes disappear sooner. However, this does not change the behavior pattern of these protocols. It is clear that AntSensNet exhibits the longest lifetime with all nodes remaining fully functional. Test results show that AntSensNet achieves more than twice the cluster head lifetime of T-ANT. That can be explained by the fact that T-ANT selects any node as a cluster head. That node can be a normal node or a resource-rich node. If a normal node is selected as CH, it must carry out some important tasks in the routing process and that implies a bigger consummation of energy. A

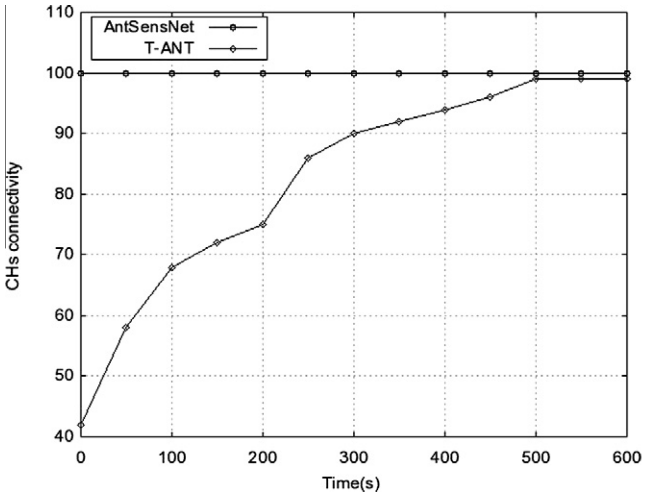


Fig. 4. The CH connectivity at various simulation time for AntSensNet and T-ANT.

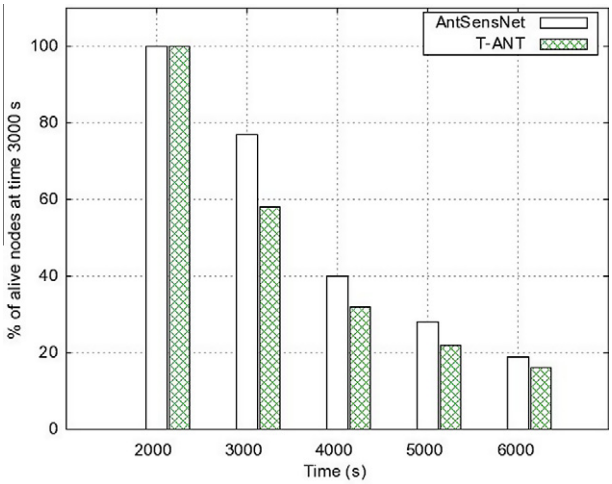


Fig. 5. Network lifetime versus simulation time for T-ANT and AntSensNet.

normal node acting as CH will deplete its battery first than a resource-rich node. AntSensNet, on the contrary, generally selects resource-rich nodes as CH. In that way the network lifetime (the time that the first node *dies*) is longer in AntSensNet than in T-ANT.

4.2. The routing process

This simulation was carried out after network clustering. The clustering parameters were $a = b = 0.5$, that is, in order to select a CH, the memory and energy components are equally important. The performance of this novel algorithm was compared with a well-known protocol, AODV, supported in NS-2. AODV was modified in order to only consider the CHs at the moment they search for network routes. This way, a version of AODV can be compared with AntSensNet. In the base scenario, 400 nodes (200 scalars and 200 multimedia) are placed in a square area of 400 m × 400 m, and $R_{cluster} = 60$ m. Simulations run for a total of 600 s every time.

Three performance metrics were taken into consideration: the *Packet Delivery Ratio* (*PDR*) involves the ratio of successfully delivered data packets to the total data packets sent from the source to their destination. *End-to-end delay* shows the amount of time needed to successfully deliver a packet from the source to the sink. *Routing overhead* indicates the ratio of routing packets transmitted to the total data packets delivered. Routing packets include control packets used for route discovery, route maintenance and pheromone updates.

To solely examine the effect of this novel routing algorithm, the network was moderately loaded. A Constant Bit Rate (CBR) traffic source model was used to generate data traffic between 32 and 1024-byte packets. Two traffic classes are produced by the nodes: multimedia traffic (with a size of 1024-byte packet) and scalar traffic (32-byte packet). Two nodes (one scalar and one multimedia) in the same cluster only send information to the sink. Obviously, multimedia traffic has higher priority than scalar traffic. Using the CBR model, the source nodes sent four data packets to the sink per second, on average.

Fig. 6 shows the Packet Delivery Ratio (PDR) of AODV, AntSensNet Scalar traffic (ASNS) and AntSensNet Multimedia traffic (ASNM). In this case, the application defined the following QoS parameters for those traffic classes:

- For ASNS: in Eq. (9), all the α values equal 0, except α_e (residual energy component), which is set to 1. The path's minimal residual energy must be superior to 0 (parameter $E_{\min}$ of Eq. (5)).
- For ASNM: in Eq. (9), all the α values are 0, except α_ε (packet loss component), which is set to 1. That is, the packet loss in the discovered routes for this traffic class must be minimal.
- Both traffic classes have the ρ values (pheromone enforcement parameter) set to 0.7 for Eqs. (10) and (11).

We find that ASNS shows a comparable average PDR with AODV, while ASNM outperforms these two protocols after a few seconds. At the beginning, ASNM lacks sufficient information in order to find appropriate routes, but after a certain period of time, when the algorithm converges and the ants have gathered much node and route information, the quality of routes discovered for the ASNM is superior to those found by ASNS and AODV. Such results were expected, and this investigation confirms the authors' hypotheses.

In Fig. 7, observe the mean end-to-end delay comparison between the protocols. For this experiment, the parameters for the ASNM traffic class were changed: all the α values are 0, except α_δ (cumulative queue delay parameter), which is set to 1. For this simulation, the

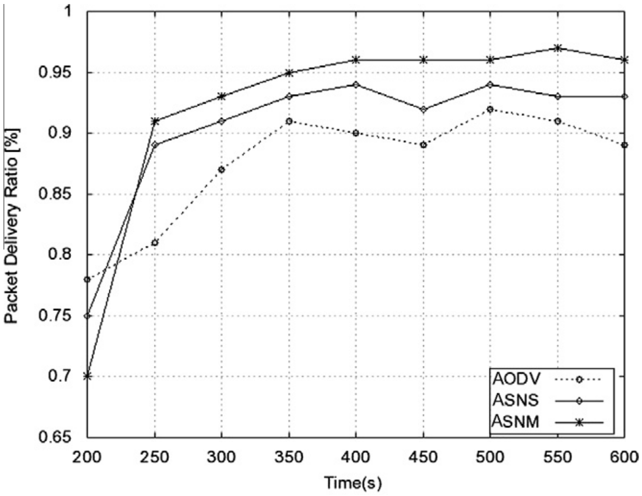


Fig. 6. Packet delivery ratio.

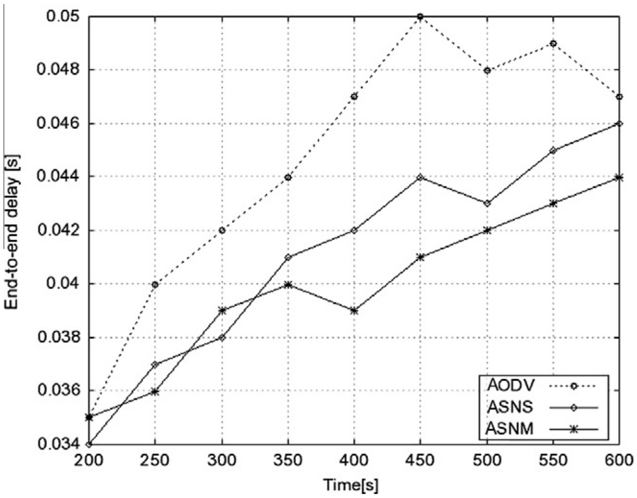


Fig. 7. End-to-end delay.

maximal delay (parameter $D_{\max}$ in Eq. (5)) is set at 8 ms. Notice that the end-to-end delays associated with the ASNM and ASNS packets are lower (and better) than those of AODV. Since multiple paths were discovered, when a path to the destination breaks, packets could immediately continue to be forwarded using another paths without a new route discovery process. Obviously, this reduced the end-to-end delay of the ASNM and ANSN packets. Since ASNS considers only energy as the main parameter for route discovery, not all packets were directed on the best path. Hence, ASNS generally requires more end-to-end delay than ASNM.

Routing overhead is shown in Fig. 8. Since extra FANT/BANT packets are required periodically to monitor and maintain path conditions, the routing overhead of AntSensNet is higher than AODV. This overhead can be reduced by embedding data into FANTs (a specimen of DANTs) and piggybacking the pheromone information on data packets if there is traffic between the sink and the CHs. Due to such periodic updates, AntSensNet constantly requires a certain amount of routing overhead.

4.3. Influence of network loading

Fig. 9 depicts the effects of network loading on the performance metrics by increasing the number of data connections from 10 to 30. These results has been gotten from the average of 10 simulations of 600 s each.

Fig. 9a shows that the PDR for the protocols AODV and AntSensNet (ASN) has a declining trend when the number of data connection (512 K CBR) is increased. The PDR of AODV dropped from 76% to 51% while that of ASN dropped from 92% to 58%. Both AODV and ASN are able to maintain >50% PDR in the latter case.

As Fig. 9b shows, for a number of data connections of 10 nodes transmitting at the same time, the protocols have relatively small latency of <20 ms. The latency increases gradually with the number of simultaneous data connections. It is worth to note that the difference in latency between the load at 30 data connections is about twice, i.e. from 75 ms for ASN and 120 ms for AODV, which is significant.

These results are explained by the AntSensNet's capability to find adapt to the network conditions, to find better routes and to use multiples routes.

4.4. Video transmission

In this section we compare the capacity of protocols ASAR [36] and TPGF [35] versus our protocol AntSensNet to transmit video packets.

The network topology used is the same used in the previous simulations. Only a video sensor in all the network is capturing, encoding and sending a live video sequences to the sink. We use only two paths to send the packets (in TPGF and AntSensNet, for ASAR we use only a path). The video sequence is encoded according to H.264/AVC standard with a reference frame list of size five frames for compensated prediction. The video testing sequence *Foreman* [39] is used at QCIF resolution with 300 video frames at a frame rate of 30 fps with a constant quantization step. In addition, the value of the parameter M_{REF} was set to five frames. The inter-frame period is 36 frames and is set to be equal with the transmission window. The video frames are encapsulated into packets of size 1024 bytes. A text trace of the video file was created including the size of each frame and the time from the start of the video that the frame occurred. This text information was included within the data field of the packets in the simulation. As each packet was sent and received, a trace file was generated indicating the time segment number of the video being sent. After the simulation, the video file was reconstructed using the Evalvid software, expanded into uncompressed video using ffmpeg [38], and compared to the original uncompressed source file again using the Evalvid software.

Fig. 10 shows the average PSNR of the *Foreman* video. We can see that the video quality was higher for the simulations using AntSensNet when compared to the other protocols. This is because the protocols TPGF and ASAR are not able to handle correctly video content.

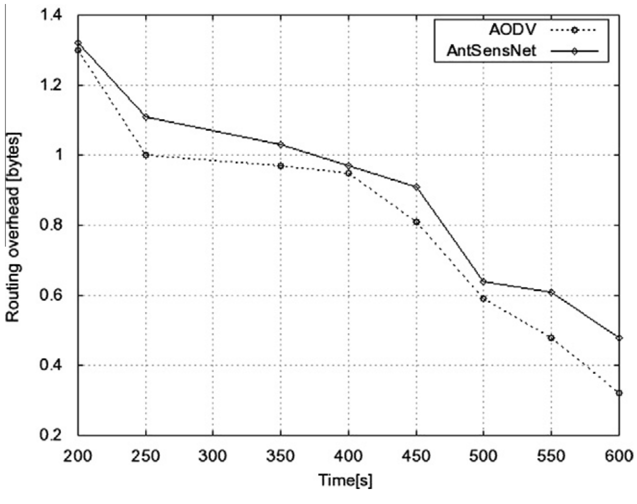


Fig. 8. Routing overhead.

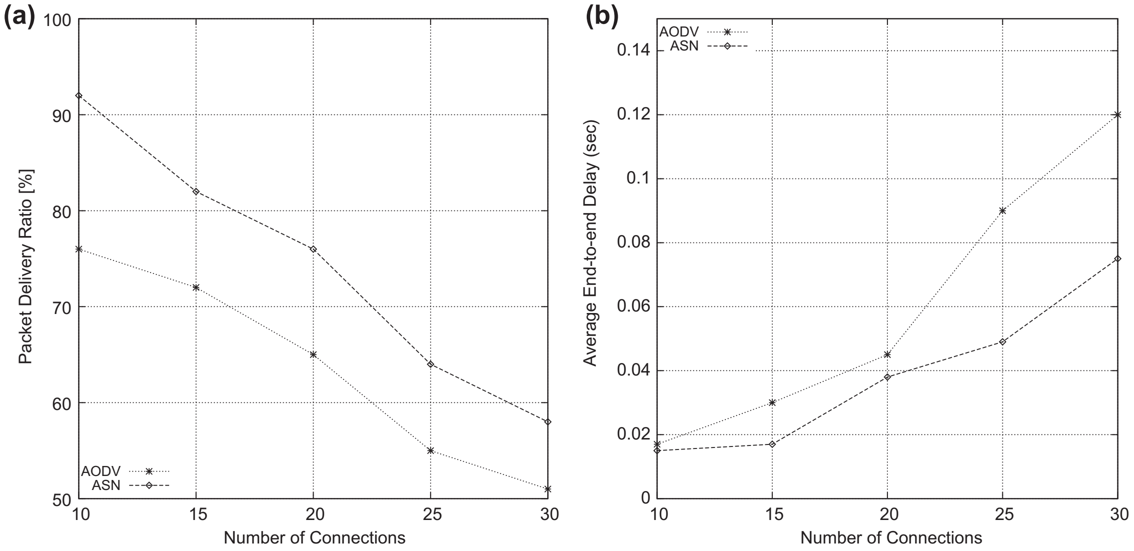


Fig. 9. Results of varying number of data connections on (a) PDR and (b) average end-to-end delay.

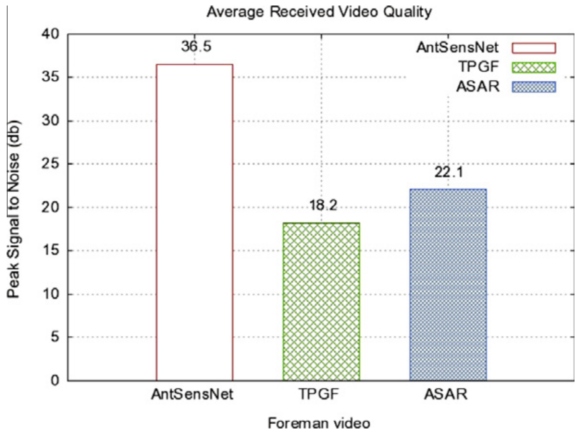


Fig. 10. Received video quality of *Foreman* video.

They do not implement any distortion minimization rate control and they are only specialized in scalar data transmission. Conversely, AntSensNet is content-aware and is able to take actions in order to minimize the video distortion.

5. Conclusion

The promising pace of technological growth has led to the design of sensors capable of sensing and producing multimedia data. However, as multimedia data contain images, video, audio and scalar data, each deserves its own metrics. These characteristics of multimedia sensor networks depend on efficient methods in order to satisfy QoS requirements. Given such motivation, this paper proposes a QoS routing algorithm such as AntSensNet for WMSNs based on an Ant Colony optimization framework and a biologically inspired clustering process. The routing algorithm also offers different classes of traffic, adapted to the needs of applications. The clustering element uses special agents (ants) to guide the selection of CHs in a totally distributed manner. In comparison with T-ANT, another ant-based clustering algorithm, this novel clustering process achieves a permanent CH connection with lower energy costs. Routing comprises both reactive and proactive components. In a reactive path setup aimed at the classes of traffic in the multimedia sensor networks, the algorithm can select paths to meet the application QoS requirements, thus improving network performance. Multimedia data are sent over the found paths. Over the course of the session, paths are continuously monitored and improved in a proactive way. Simulation results show that the performance of AntSensNet outperforms the standard AODV in terms of delivery ratio, end-to-end delay and routing overhead.

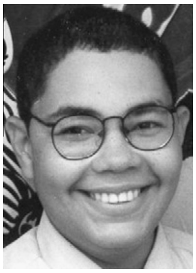
Simulation results support that the proposed distortion reduction mechanism used to transport video packets results in better quality video than using other protocols for multimedia transport (TPGF and ASAR).

In future work, we intend to study the initialization method to populate routing tables with initial pheromone levels. As shown in the literature [33], such mechanisms can further increase network efficiency. Other approaches to be studied include the integration of multiple sink-nodes as well as node mobility. Another improvement we plan to investigate is to extend the proposed architecture to a t cross-layer architecture proposed in this work to include a better interaction with a transport entity and the MAC sublayer. Similarly, Instead of the 802.11 MAC layer, we will investigate the use of Sensor MAC (SMAC) which is a MAC protocol designed for wireless sensor networks. SMAC has the potential to make the cross-layer architecture more energy efficient.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Computer Networks 38 (4) (2002) 393–422.

[2] I.F. Akyildiz, T. Melodia, K.R. Chowdhury, A survey on wireless multimedia sensor networks, Computer Networks 51 (4) (2007) 921–960.
[3] K. Akkaya, M. Younis, A survey on routing protocols for wireless sensor networks, Ad Hoc Networks 3 (3) (2005) 325–349.
[4] W.-M. Chen, C.-S. Li, F.-Y. Chiang, H.-C. Chao, Jumping ant routing algorithm for sensor networks, Computer Communications 30 (14–15) (2007) 2892–2903.
[5] R. GhasemAghaei, A. Rahman, W. Gueaieb, A.E. Saddik, Ant colony-based reinforcement learning algorithm for routing in wireless sensor networks, in: IEEE Instrumentation and Measurement Technology Conference Proceedings, 2007, IMTC 2007, Warsaw, Poland, 2007, pp. 1–6.
[6] R. Muraleedharan, L.A. Osadciw, A predictive sensor network using ant system, in: SPIE Defense and Security Symposium, Proceedings of the SPIE – The International Society for Optical Engineering, vol. 5440, SPIE – Int. Soc. Opt. Eng., Orlando, FL, USA, 2004, pp. 181–192.
[7] Y. Zhang, L.D. Kuhn, M.P.J. Fromherz, Improvements on ant routing for sensor networks, in: Ant Colony, Optimization and Swarm Intelligence, Lecture Notes in Computer Science, first ed., vol. 3172, Springer, Berlin, 2004, pp. 154–165.
[8] M. Liu, Y. Sun, R. Liu, X. Huang, An improved ant colony QoS routing algorithm applied to mobile ad hoc networks, in: International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM'07, Shanghai, China, 2007, pp. 1641–1644.
[9] M. Dorigo, T. Sttzle, Ant Colony Optimization, MIT Press, Cambridge, Mass, 2004.
[10] M. Dorigo, G.D. Caro, Antnet: distributed stigmergetic control for communications networks, Journal of Artificial Intelligence Research 9 (1998) 317–365.
[11] K. Sohrabi, J. Gao, V. Ailawadhi, G.J. Pottie, Protocols for self-organization of a wireless sensor network, IEEE Personal Communications 7 (5) (2000) 16–27.
[12] K. Akkaya, M. Younis, Energy and QoS aware routing in wireless sensor networks, Cluster Computing 8 (2–3) (2005) 179–188.
[13] L. Chenyang, B.M. Blum, T.F. Abdelzaher, J.A. Stankovic, H. Tian, RAP: a real-time communication architecture for large-scale wireless sensor networks, in: Eighth IEEE Real-Time and Embedded Technology and Applications Symposium, San Jose, CA, USA, 2002, pp. 55–66.
[14] T. He, J.A. Stankovic, C. Lu, T. Abdelzaher, SPEED: a stateless protocol for real-time communication in sensor networks, in: 23rd IEEE International Conference on Distributed Computing Systems (ICDCS-03), Providence, RI, United States, 2003, pp. 46–55.
[15] D.P. Agrawal, A. Mahapatra, K. Anand, QoS and energy aware routing for real-time traffic in wireless sensor networks, Computer Communications 29 (4) (2006) 437–445.
[16] E. Felemban, L. Chang-Gun, E. Ekici, MMSPEED: multipath multi-speed protocol for QoS guarantee of reliability and timeliness in wireless sensor networks, IEEE Transactions on Mobile Computing 5 (6) (2006) 738–754.
[17] B. Deb, S. Bhatnagar, B. Nath, ReInforM: reliable information forwarding using multiple paths in sensor networks, in: 28th Annual IEEE International Conference on Local Computer Networks, Los Alamitos, CA, USA, 2003, pp. 406–415.
[18] M. Dorigo, C. Blum, Ant colony optimization theory: a survey, Theoretical Computer Science 344 (2–3) (2005) 243–278.
[19] M. Dorigo, L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, IEEE Transactions on Evolutionary Computation 1 (1) (1997) 53–66.
[20] B. Bullnheimer, R.F. Hartl, C. Strauss, An improved ant system algorithm for the vehicle routing problem, Annals of Operations Research 89 (1999) 319–328.
[21] L. Rosati, M. Berioli, G. Reali, On ant routing algorithms in ad hoc networks with critical connectivity, Ad Hoc Networks 6 (6) (2007) 827–859.
[22] M. Gunes, U. Sorges, I. Bouazizi, ARA-the ant-colony based routing algorithm for manets, in: Proceedings of International Conference on Parallel Processing Workshops, IEEE, Vancouver, Canada, 2002, pp. 79–85.
[23] M. Gerla, K. Xu, Multimedia streaming in large-scale sensor networks with mobile swarms, ACM SIGMOD Record 32 (4) (2003) 72–76.
[24] S. Ziane, A. Melou, A swarm intelligent multi-path routing for multimedia traffic over mobile ad hoc networks, 1089771 55-62, October 2005.
[25] M.A. Rahman, R. GhasemAghaei, A.E. Saddik, W. Gueaieb, M-IAR: biologically inspired routing protocol for wireless multimedia sensor networks, in: IEEE Instrumentation and Measurement Technology Conference Proceedings, Victoria, BC, 2008, pp. 1823–1827.
[26] G. Feng, C. Douligeris, K. Makki, N. Pissinou, Performance evaluation of delay-constrained least-cost QoS routing algorithms based on linear and nonlinear lagrange relaxation, in: 2002 IEEE International Conference on Communications, ICC 2002, vol. 4, Piscataway, NJ, USA, 2002, pp. 2273–2278.
[27] W.B. Heinzelman, A.P. Chandrakasan, H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, IEEE Transactions on Wireless Communications 1 (4) (2002) 660–670.
[28] S. Selvakennedy, S. Sinnappan, Y. Shang, A biologically-inspired clustering protocol for wireless sensor networks, Computer Communications 30 (14–15) (2007) 2786–2801.
[29] L.M.C. Arboleda, N. Nasser, Comparison of clustering algorithms and protocols for wireless sensor networks, in: Canadian Conference on Electrical and Computer Engineering, Canadian Conference on Electrical and Computer Engineering, CCECE'2006, 2006, pp. 1787–1792.
[30] S. Misra, M. Reisslein, X. Guoliang, A survey of multimedia streaming in wireless sensor networks, IEEE Communications Surveys and Tutorials 10 (4) (2008) 18–39.
[31] VINT, The network simulator – ns-2, 2008, <http://www.isi.edu/nsnam/ns/>.
[32] O. Younis, S. Fahmy, HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks, IEEE Transactions on Mobile Computing 3 (4) (2004) 366–379.
[33] Z. Ying, L.D. Kuhn, M.P.J. Fromherz, Improvements on Ant Routing for Sensor Networks, Springer-Verlag, 2004.
[34] T. Dagiuklas, S. Kotsopoulos, I. Politis, M. Tsagkaropoulos, Power efficient video multipath transmission over wireless multimedia sensor networks, Mobile Networks and Applications 13 (3) (2008) 274–284.
[35] L. Shu, Y. Zhang, L.T. Yang, Y. Wang, M. Hauswirth, N. Xiong, TPGF: geographic routing in wireless multimedia sensor networks, Telecommunication Systems (Nov.) (2009).
[36] Y. Sun, H. Ma, L. Liu, Y. Zheng, ASAR: an ant-based service-aware routing algorithm for multimedia sensor networks, Frontiers of Electrical and Electronic Engineering in China 3 (1) (2008) 25–33.
[37] L. Golubchik, J.C.S. Lui, T.F. Tung, A.L.H. Chow, W.J. Lee, G. Franceschinis, C. Anglano, Multi-path continuous media streaming: what are the benefits?, Performance Evaluation, Netherlands (2002) 429–449
[38] F. Bellard, <http://www.ffmpeg.org>.
[39] Arizona State University Video Traces Research Group, <http://traces.eas.asu.edu/yuv/index.html>.



Luis Cobo received the Engineer Degree and the Master's Degree in Computer Engineering from University of the Andes, Bogota, Colombia, in 1995 and 1997, respectively. He is currently working toward the Ph.D. Degree in Computer Engineering at École Polytecnique de Montréal, Canada. He is working in biologically inspired routing for wireless multimedia sensor networks.

Alejandro Quintero received the engineer degree in computer engineering from Los Andes, Colombia, in 1983. In June 1989, and in 1993, he received the diploma of advanced studies, and the Ph.D. degree in computer engineering, respectively, from the INPG – Grenoble and Joseph Fourier University, Grenoble, France. He is currently a full professor at the Department of Computer Engineering of École Polytechnique de Montreal, Canada. His main research interests include services and applications related to mobile computing, networks infrastructures and next generation mobile networks. Dr. Quintero is the coauthor of one book, as well as over 50 other technical publications including journal and proceedings papers.

Samuel Pierre received the B.Eng. degree in civil engineering in 1981 from École Polytechnique de Montréal, Québec, the B.Sc. and M.Sc. degrees in mathematics and computer science in 1984 and 1985, respectively, from the UQAM, Montréal, the M.Sc. degree in economics in 1987 from the Université de Montréal, and the Ph.D. degree in Electrical Engineering in 1991 from École Polytechnique de Montréal. He is currently a Professor of Computer Engineering at École Polytechnique de Montréal where he is Director of the Mobile Computing and Networking Research Laboratory (LARIM), chairholder of the NSERC - ERICSSON Chair in Next Generations Mobile Networking Systems and Director of Mobile Computing and Networking Research Group (GRIM). His research interests include wireline and wireless networks, mobile computing, artificial intelligence, and telelearning. He is a senior member of IEEE and a member of ACM. He is a Regional Editor of the Journal of Computer Science, an Associate Editor of IEEE Communications Letters, IEEE Canadian Journal of Electrical and Computer Engineering and IEEE Canadian Review, and serves on the editorial board of Telematics and Informatics edited by Elsevier Science. He has received many distinctions, such as the Prix Poly 1873 for excellence in teaching (2001 and 2005), Fellow of Engineering Institute of Canada (2003), among others. He is a Fellow of the Canadian Academy of Engineering (2008). In 2009, he has been awarded Chevalier de l'Ordre national du Québec, ''the highest distinction given by the Québec government.''