Performance Evaluation of the DAlPaS Congestion Control Algorithm in Wireless Sensor Networks

Charalambos Sergiou and Vasos Vassiliou
Networks Research Laboratory
Department of Computer Science
University of Cyprus
Nicosia, Cyprus
Email: {sergiou,vasosv}@cs.ucy.ac.cy

***Abstract*—As wireless sensor networks are evolving to applications where high load demands dominate and time is considered critical, congestion remains a serious problem that has to be effectively and efficiently tackled. Especially in critical applications such as industrial or aviation control, congestion occurrence, in any instance of the network's operation, is capable of ruining the mission of the network. Congestion is usually controlled by reducing the rate with which sources are injecting data in the network ("traffic control"). Although this method is effective in many cases, it is not acceptable for applications that need all the produced data to be received by sink. Besides the "traffic control" method, "resource control" is also used for congestion mitigation. Algorithms that employ the "resource control" method employ alternative paths toward the sink. In this work, through analysis and simulations, we evaluate the performance of Dynamic Alternative Path Selection Algorithm (DAlPaS) a congestion control algorithm that employs alternative paths for the transmission of excess packets from the source(s) to the sink(s). The DAlPaS algorithm is evaluated against two other algorithms that also employ "resource control" (TARA and HTAP) and against an algorithm that employs the "traffic control" method (SenTCP), in addition to the no congestion control case (no CC).**

***Keywords*-Wireless Sensor Networks, Multipath Routing, Congestion Control, Energy Utilization**

I. Introduction

Wireless Sensor Networks (WSNs) are ad hoc networks composed of sensor nodes, deployed in various fields, capable of sensing various phenomena. Upon the appearance of the monitored event, sensor nodes transform the analog data that they sense to digital and transmit them to destination nodes (usually called sinks). Wireless sensor nodes are battery powered nodes. Due to severe power limitations that derives from battery-based operation, their communication and processing abilities are also limited. Thus, for the transmission of data packets from the sources (the nodes that sense the phenomenon) to the sinks (the final destination nodes) nodes cooperate and transmit their packets in a hop-by-hop manner. Frequently, sensor nodes are densely deployed near the possible event sources and sinks in a redundant manner [1] [2]. WSNs comprise of a potential large set of nodes that may be distributed over a wide geographical area indoor or outdoor. Traffic patterns in WSNs can be derived from the physical processes that they sense. WSNs typically operate under light load and suddenly become active in response to a detected or monitored event. When the event occurs, it is normally sensed by a relatively big number of nodes which are becoming the sources. These source nodes start injecting packets into the network until the event stops. Since the number of sources is usually more than one, there is a big possibility for congestion to occur in the network. Congestion can occur either in the medium (packet collisions), if a contention based MAC protocol is used, or in a node's buffer, if the node receives packets at a higher rate than it can transmit. Currently there are two major methods used in congestion control algorithms in order to face congestion in WSNs: "traffic control" and "resource control". By employing the "traffic control" method, source nodes are forced to reduce the rate with which they are injecting packets in the network, until the load in the network is aligned with the capacity of the paths that are being used. On the other hand, by using the "resource control" method, the source data rate remains constant and the capacity of the network is adapted to the load requirements by employing nodes which are not in the initial paths from the source(s) to the sink(s). In this paper we provide a concise, but comprehensive, analysis of the energy efficiency of the two categories of algorithms and evaluate the performance of the "Dynamic Alternative Path Selection" (DAlPaS) algorithm [3] a congestion control algorithm in WSNs that bases its functionality on a "resource control" method. The performance of DAlPaS is compared against two other algorithms that employ "resource control" method for congestion control, TARA [4] and HTAP [5], an algorithm that employs "traffic control" method, SenTCP [6], as well as the case where no Congestion Control algorithm is employed (no CC).

II. Analysis

We consider an event-driven hierarchical network where all nodes forward their packets upstream to the sink. Nodes are deployed on the crossed lines of a grid and each node transmits packets only one hop away. In this network there is only one sink. Each node has a sensing range of R meters and a communication range of 2R meters. We also consider the occurrence of a permanent event in the network which is captured by the nodes which are in R meters distance from the

event. These nodes are becoming source nodes and attempt to forward their packets to the sink. This network presents, in our opinion, the "ideal" setup for congestion occurrence.

A. Energy Efficiency Analysis

It is known from [7] that the energy consumed when sending a packet of m bits over a one hop wireless link of distance d, can be expressed as:

$$E_h(m,d) = \{E_T(m,d) + P_T T_{st} + E_{encode}\} + \{E_R(m) + P_R T_{st} + E_{decode}\} \quad (1)$$

where

E_T = energy used by the transmitter circuitry and power amplifier

E_R = energy used by the receiver circuitry

P_T = power consumption of the transmitter circuitry

P_R = power consumption of the receiver circuitry

T_{st} = startup time of the transceiver

E_{encode} = energy used to encode

E_{decode} = energy used to decode

If we consider that there is a path of n nodes between a source node and the sink, the energy consumed for transmitting this packet over this path is

$$E_{path} = E_T(m,d) + \left\{ \sum_{i=2}^{n} E_T(m,d) + E_R(m) \right\} + \beta, \quad (2)$$

where β is the sum of P_T, P_R, T_{st}, E_{encode} and E_{decode} for each node.

Using these equations we can claim that each node that is not going to become a source node during its lifetime in WSNs, is able to relay a finite maximum number of packets. The number of packets is strictly related to its initial energy and the distance d of its next-hop neighbor.

At this time we define two more variables. Sensor Lifetime SL is the total time that a node remains alive in the network and it depends on its energy consumption per unit time. So

$$SL = \frac{E_0}{E_t}, \quad (3)$$

where E_0 is the initial energy of the node and E_t the energy consumption per time unit. Energy consumption per time unit is strictly depended on the rate of packets that it receives and transmits and the distance over which it transmits them.

On the other hand, Network Lifetime NL is the total time that elapses until the network is not able to transmit packets from the source(s) to the sink(s) anymore.

Without loss of generality we consider a grid placement where all nodes are equally spaced and each node transmits upstream (to sink direction) only one hop away. In this case since distance d is constant and the size of packet is also the same (m), we can safely denote that the only major variable factor that affects the lifetime of a relay node in this case, is the number of packets that it transmits and receives.

B. "Resource" vs "Traffic" Control

For our analysis we consider that the network, upon its configuration, is facing an event that creates a heavy data load. In this case congestion control algorithms need to be applied. If no congestion control algorithm is applied we can calculate from Equations 1 and 2 that the network's lifetime is very limited. If we consider that each node has a buffer capacity of C packets of m bits each, and each node is receiving packets with a higher rate than it is capable of transmitting, then this buffer will soon become full. The furthest away from the source this node is, the more energy is going to be wasted (network-wide) when a packet drop occurs, according to Equation 2.

Therefore, it is evident that congestion control algorithms need to be involved. As we discussed before, currently there are two types of congestion control algorithms: "Traffic Control" and "Resource Control". Studying these two methods, from the energy consumption perspective, we can deduce the following: Algorithms that employ the "traffic control" method, although they reduce the load in the network in order to avoid congestion they do not alter the routing path and keep transmitting packets from the same path. In such case the energy consumption per time unit (E_t) of these nodes is increasing and the sensor node lifetime SL is reducing. So if we consider that each node in this path is relaying p packets/s, then combining Equations 1 and 3 we can derive that each node in this path consumes $p \times E_h(m,d)$ Joule/s. As time is increasing, E_t is increasing as well, and according to Equation 3, SL reduces. Since this situation is the same for the whole path, it is expected that in finite time t, which is heavily depended on the duration that this path is continually used, the node will be power exhausted. When nodes are power exhausted, the network's lifetime is decreasing even if the energy of the nodes in the rest of the network remains unaltered.

When the "resource control" method is employed the situation is different. In this case, when a node becomes a hotspot (buffer or medium congested) the excess packets are routed through other nodes. In addition, since algorithms that employ this method always start using the shortest path, it means that in the case of congestion, extra packets are routed through longer routes. The equation which is actually affected in this method is Equation 2. In this case, parameter n, which is the number of nodes that form the path, is increasing, leading to higher energy consumption. If we consider that algorithms using this method always employ a topology control scheme [4], [5], the number of alternative paths is increased but the nodes that implement these paths are limited in order to avoid long routes. This means that several paths can be used, leading to balanced energy consumption. Therefore, the network is exhausting its energy uniformly and its Network Lifetime is increased.

If we compare the two methods after an intensive congestion situation of duration τ it is possible to have the following "strange" relationship between "traffic control (tc)" and "re-

source control (rc)" concerning their total remaining energy :

$$\sum_{i=1}^{m} E_{tc} \geq \sum_{i=1}^{m} E_{rc} \text{ and } NL_{tc} < NL_{rc} \quad (4)$$

where m is the initial number of nodes in the network.

This means that it is possible for the sum of nodes remaining power to be bigger in the "traffic control" method compared to the "resource control" method, but at the same time the network's lifetime to be smaller. It is clear that the maximum lifetime can be achieved when nodes are exhausting their power uniformly.

III. Description of Employed Algorithms

DAlPaS [3] is a congestion control and avoidance algorithm that attempts to choose an alternate path in case of congestion taking into account a number of basic performance parameters. Complementary to Energy Aware Protocols [8][9] that find the lowest energy route or energy sufficient paths to forward data and base their path alternation decision on these conditions, DAlPaS also takes into consideration the node's congestion situation (both in terms of buffer occupancy and channel interference). On the other hand, while congestion control and reliable data transmission protocols like [4] [5][10] base their "alternate path" decision on a congestion threshold or the path's cost, DAlPaS also counts the node's remaining power. DAlPaS is completely dynamic and distributed algorithm.

In first place implements two stages. A soft and a hard stage. When soft stage applies, a node that receives packets from more than one flows keeps servicing the flow from which it receives packets with the higher rate and informs the nodes from which the other flows are coming to change destination node. Using this proactive method, network, avoids in first place possible hotspots especially when the load is not so big in network (transient conditions). On the other hand, nodes enter hard stage when they must prohibit flows from reaching them. In this case a "Flag Decision Mechanism" is activated. Flag decision algorithm runs when a node enters in hard stage. In this stage a node becomes temporarily or permanently unable to accept any more packets from any flows. A node may become unable to receive data for the following reasons:

- Buffer Occupancy is reaching its upper limit
- Low Remaining Power
- Higher level node unavailability

TARA (Topology Aware Resource Adaptation) [4] protocol focuses on the adaptation of network's extra recourses in case of congestion, alleviating intersection hot spots. TARA copes with buffer occupancy as well as channel loading. In TARA, congestion alleviation is performed with the assistance of two important nodes. These are the distributor and the merger nodes. Between them a "detour path" is established starting at the distributor and ending at the merger. The distributor distributes the traffic coming from the hot spot between the original path and the detour path, while the merger merges the two flows. Thus, in case of congestion and creation of hot- spot, traffic is deflected from the hot spot through the distributor node along the detour and reaches the merge node, where the flows are merged. As long as congestion has been alleviated the network stops using the detour path. For quick adaptation the distributor node keeps in its memory which neighbor is on the original path.

HTAP [5] is scalable and distributed framework for minimizing congestion and assuring reliable data transmissions in event based networks. As such it does not employ rate limiting actions, but tries to maintain a high level of packet rate while minimizing packet losses. It is based on the creation of alternative paths from the source to sink, using the plethora of a networks unused nodes, in order to safely transmit the observed data. The creation of alternative paths involves several nodes which are not in the initial shortest path from the source to the sink. The use of these nodes leads to a balanced energy consumption, avoiding the creation of holes in the network and prolonging network lifetime. The HTAP algorithm consists of four major parts.

- Flooding with level discovery functionality: Through this procedure, each node discovers its neighbor nodes and updates its neighbor table. In addition, sensor nodes are placed in levels from the source to the sink.
- Alternative Path Creation Algorithm: In order to avoid congestion each candidate congested receiver is sending a backpressure packet to the sender. So the sender stops the transmission of packets to the candidate congested receiver and searches in its neighbor table to find the least congested receiver in order to continue the transmission of data. The dynamic change of the receivers leads to the creation of new routes from the source to the sink.
- The Hierarchical Tree Algorithm: A hierarchical tree is created beginning at the source node. Connection is established between each transmitter and receiver using a 2-way handshake. Through this packet exchange, the congestion state of each receiver is communicated to the transmitter. The combination of the two algorithms implements Hierarchical Tree Alternative Path (HTAP) algorithm. Specifically when the neighbor nodes of a specific node is below a specified threshold the APC algorithm applies, the HT applies otherwise
- Handling of Powerless (Dead Nodes): Special care is taken in the HTAP algorithm concerning the nodes which their battery is exhausted. Thus, when a node is going to lose its power, it is immediately extracted from the network and the tables of its neighbor nodes are updated.

SenTCP [6] SenTCP is an open-loop hop-by-hop congestion control protocol with two special features:

1) It jointly uses average local packet service time and average local packet inter-arrival time in order to estimate current local congestion degree in each intermediate sensor node. The use of packet arrival time and service

time not only precisely calculates congestion degree, but effectively helps to differentiate the reason of packet loss occurrence in wireless environments, since arrival time (or service time) may become small (or large) if congestion occurs.

2) It uses hop-by-hop congestion control. In SenTCP, each intermediate sensor node will issue feedback signal backward and hop-by-hop. The feedback signal, which carries local congestion degree and the buffer occupancy ratio, is used for the neighboring sensor nodes to adjust their sending rate in the transport layer. The use of hop-by-hop feedback control can remove congestion quickly and reduce packet dropping, which in turn conserves energy.

IV. Algorithm Evaluation

The DAlPaS algorithm was evaluated through simulations and its performance is presented in this section, along with the performance of four other algorithms described in Section III. The evaluation was performed using Prowler, a probabilistic wireless network simulator [11]. Prowler provides a radio fading model with packet collisions, static and dynamic asymmetric links, and a CSMA MAC layer.

A. Simulation Environment

To perform our simulations we have used the radio propagation model provided by Prowler. The transmission model is given by:

$$P_{rec,ideal}(d) \leftarrow P_{transmit}\frac{1}{1+d^{\gamma}} \tag{5}$$

where, $2 \leq \gamma \leq 4$ and

$$P_{rec}(i,j) \leftarrow P_{rec,ideal}(d_{i,j})(1+a(i,j))(1+\beta(t)) \tag{6}$$

where $P_{transmit}$ is the signal strength at the transmitter and $P_{rec,ideal}(d)$ is the ideal received signal strength at distance d, a and β are random variables with normal distributions $N(0,\sigma_a)$ and $N(0,\sigma_\beta)$, respectively. A node j can receive packets from node i if $P_{rec}(i,j) > \Delta$ where Δ is the threshold.

B. Simulator Setup

In our simulations we set $\sigma_a = 0.5$, $\sigma_\beta = 0.03$ and $p_{error} = 0.05$. The reception threshold is set to be $\Delta = 0.1$. These are the default parameters of the simulator.

The rest of the parameters we employ represent those of the Mica-Z node and the most important of them are presented in Table I.

For each of the performance metrics presented below, the results is the average of 20 runs for 10s for each measurement point (except for the cases that it is otherwise stated). Nodes are placed uniformly on a square grid over a 500x500m area. The initial number of deployed nodes is 100. The event happens randomly in the left bottom quadrant of the grid, while the sink is situated on the upper right edge of the grid. Nodes have a sensing range of 25 meters and a communication range of 50 meters. All nodes that sense the event are becoming source nodes using the rest as relay nodes to the sink.

TABLE I
Simulation Parameters

Max Data Rate (kbps)	250
Transmission Power (dbm)	2
Receive Threshold (dbm)	-74
Transmission Current (mA)	17.4
Receive Current (mA)	19.7
Fragment Size (bit)	1024
Buffer Size(Bytes)	512K
MAC layer	CSMA/CA

C. Performance Metrics

The first performance metric we have examined is the percentage of successfully received packets (Eq. 7).

$$ReceivedPktsRatio(\%) = \frac{SuccessfullyReceivedPkts}{TotalPktsSent} \tag{7}$$

This metric is particularly important for critical/emergency applications, where every packet has to be received by the sink and indicates the ability of algorithms to maintain a robust network. Results for this particular metric are presented in Figure 1.

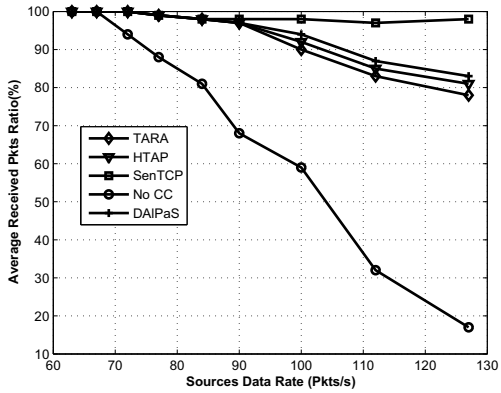

Fig. 1. Percentage of Successfully Received Packets

It is clear that if no congestion control algorithm is applied in a WSN while the amount of data injected in the network is increasing, this will result in the severe degradation of the network's performance, since an important percentage of the transmitted packets will not reach the sink. On the other hand, when congestion control algorithms are applied the results are significantly improved. Concerning congestion control algorithms, we notice that the traffic-control SenTCP algorithm has the best behavior, with respect to packet loss, since it controls the rate with which sources are injecting packets in the network, so it optimizes the performance of the network. Resource control algorithms start degrading (i.e. start

having packet losses) when a large number of data packets are injected in the network (>90 pkts/s in this setup). The reason for this attitude lies on the fact that at this data rate the network's resources are fully utilized after a specific period of time and "resource control" cannot efficiently continue to be applied. Irrespective of that, if we compare the performance of the three "resource control" algorithms we notice that the DAlPaS algorithm presents a significantly better performance than the others.

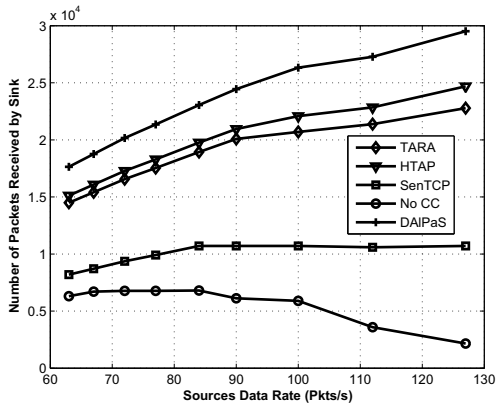

Fig. 2. Number of Packets received by the Sink

The advantage of DAlPaS is more evident if we consider the total number of packets that manage to reach the sink (see Figure 2). DAlPaS manages to deliver 20-25% more packets to the sink, compared to HTAP and TARA respectively, and three times as many packets from SenTCP. This benefit is attributed to the employment of a "soft stage topology control" where nodes are being "advised", in the first place, to avoid using as next nodes, nodes that already serve another flows. This fact helps the network to utilize its resources more uniformly.

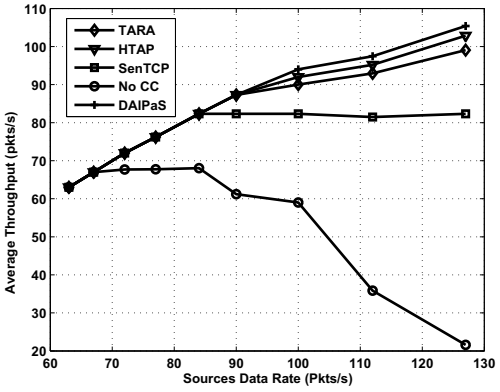

Fig. 3. Average Throughput with increasing load

The next parameter we have analyzed is the rate of data that reach the sink (throughput). The graph in Figure 3 should be considered in conjunction with Figure 1 in order to compare the actual number of packets that reach the sink when each algorithm is applied. We recognize that the DAlPaS algorithm exhibits the best performance. It is important to recognize that, as far as this metric is concerned, the "resource control" algorithms have an advantage over their "traffic control" counterparts. While SenTCP limits its sending (and receiving) rate per node at an average of 82 Pkts/sec, the throughput for DAlPaS, TARA, and HTAP keeps increasing as the source(s) send more data.

The ability of resource-control algorithms to control congestion depends on the available unused resources that can be employed to form alternative paths. To confirm the reasoning presented above concerning their performance degradation (with respect to Packet Loss), we kept the source data rate stable at 128 packets/s (the maximum rate used) while increasing the number of nodes from 100 to 500.

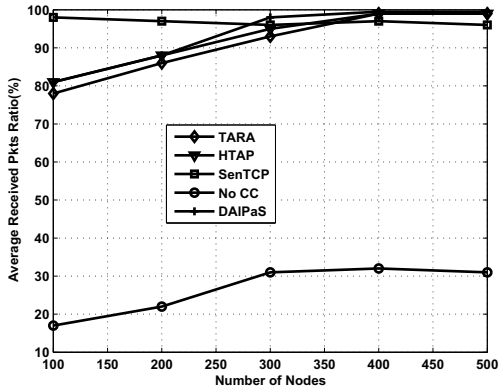

Fig. 4. Average Received Packets Ratio (%) with increasing number of nodes

In Figure 4 we notice that as the number of nodes is increasing, "resource control" algorithms are able to deliver all packets to the sink. In such a case they are able to find the resources that they need in order to successfully direct all packets to the sink without packet drops. Again if we compare the three "resource control" algorithms between them, we notice that DAlPaS presents the best performance since it starts delivering more packets to the sink, with less resources, in comparison with TARA and HTAP.

The next metric we have evaluated is the "average hop-by-hop delay". This metric is an indication of how algorithms handle inter-path interferences, link layer retransmissions and overhead introduced by control packet exchanges.

The best performance in this metric is presented by the SenTCP algorithm. This is normal, since after a congestion occurrence, the data rate in the network is reduced, packet drops are also reduced, and retransmissions and control packets are minimized. Concerning the three resource-control algorithms, the situation is the same as before. Again, DAlPaS presents the least delay between the three algorithms. It manages that because its "flag detection mechanism" spreads the traffic in

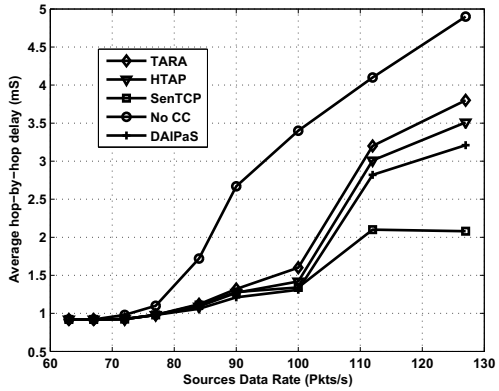

Fig. 5. Average hop-by-hop delay

a better way and helps keeping the queueing and processing delays low in the network.

Finally we evaluate the percentage of the network's remaining energy. In this simulation series opposed to the other metrics, simulations run until the network was not able to deliver packets from the source to the sink, due to a disconnected network caused by power exhausted nodes. Calculating the remaining power of "alive" nodes we get an indication of which algorithm manages to utilize more efficiently the network resources and increase the network's lifetime. In this series of simulations we used 300 nodes, with a source data rate of 100 packets/s.

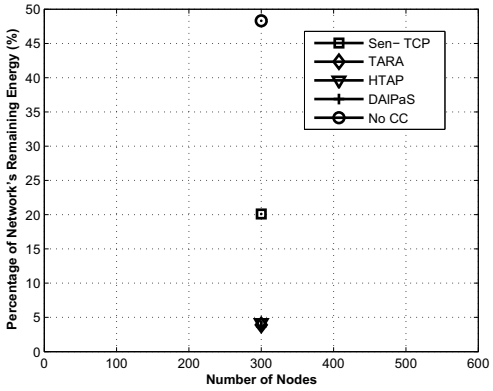

Fig. 6. Percentage of Networks Remaining Energy

Figure 6 indicates that when SenTCP is employed in the network, the network "stalls" when the total remaining power of nodes is near 20%. In other words, when the network utilizes 80% of its resources it stops performing. On the other hand "resource control" algorithms manage to utilize almost all network resources (more than 95%). This is a strong indication that nodes are exhausting their power uniformly and the network's lifetime extends considerably.

If we focus on the three resource control algorithms (Figure 7) we notice that, again, the DAlPaS algorithm presents better results than HTAP and TARA.

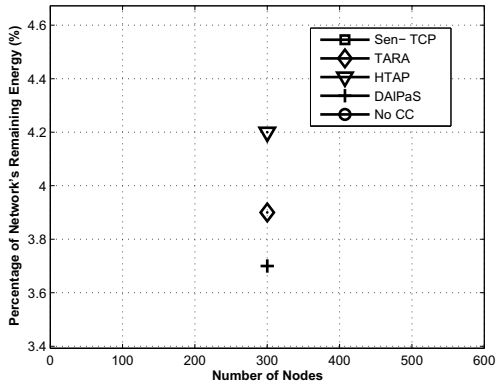

Fig. 7. Percentage of Networks Remaining Energy (zoom of lower part of Figure 6)

Figures 6 and 7 confirm the findings of Equation 4 in Section II

V. Conclusions

In this paper we studied the performance of the DAlPaS algorithm, an algorithm designed for congestion control and avoidance in Wireless Sensor Networks. DAlPas is a "resource control" congestion control algorithm and it was evaluated against TARA and HTAP which are also "resource control" algorithms, as well as against "SenTCP" which is a "traffic control" algorithm and "No CC", which represents the case where no congestion algorithm is applied in the network. Both theoretical analysis and simulation results prove that "resource control" algorithms have a better lifetime than "traffic control" algorithms. The DAlPaS algorithm was found to outperform the other two "resource control" algorithms (TARA and HTAP). Also in comparison with "SenTCP", which is a "traffic control" algorithm, the results for DAlPaS are better under specific circumstances, which are analyzed in the paper.

Acknowledgments.

This work has been conducted under the European Union Project GINSENG funded under the FP7 Program (FP7/2007-2013) grant agreement no 224282.

References

[1] V. Vassiliou and C. Sergiou, "Performance Study of Node Placement for Congestion Control in Wireless Sensor Networks," in *NTMS*. IEEE, 2009, pp. 1–8.

[2] S. Toumpis and L. Tassiulas, "Optimal Deployment of Large Wireless Sensor Networks," *IEEE Transactions on Information Theory*, vol. 52, no. 7, pp. 2935–2953, 2006.

[3] C.Sergiou and V. Vassiliou, "DAlPaS: A Performance Aware Congestion Control Algorithm in Wireless Sensor Networks," in *18th International Conference on Telecommunications (ICT 2011)*, 8-11, May 2011, pp. 178–184.

[4] J. Kang, Y. Zhang, and B. Nath, "TARA: Topology-Aware Resource Adaptation to Alleviate Congestion in Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 7, pp. 919–931, 2007.

[5] C. Sergiou, V. Vassiliou, and A. Pitsillides, "Reliable Data Transmission in Event-Based Sensor Networks During Overload Situation," in *WICON '07: Proceedings of the 3rd International Conference on Wireless Internet*, Austin, Texas, October 2007, pp. 1–8.

[6] C. Wang, K. Sohraby, and B. Li, "SenTCP: A Hop-by-Hop Congestion Control Protocol for Wireless Sensor Networks," *IEEE INFOCOM (Poster Paper)*, March 2005.

[7] H. Karvonen, Z. Shelby, and C. Pomalaza-Raez, "Coding for Energy Efficient Wireless Embedded Networks," in *Wireless Ad-Hoc Networks, 2004 International Workshop on*, 2004, pp. 300 – 304.

[8] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-Resilient, Energy-Efficient Multipath Routing in Wireless Sensor Networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 5, no. 4, pp. 11–25, 2001.

[9] R. P. Mann, K. R. Namuduri, and R. Pendse, "Energy-Aware Routing Protocol for Ad Hoc Wireless Sensor Networks," *EURASIP Journal Wireless Communications Networks*, vol. 2005, no. 5, pp. 635–644, 2005.

[10] F. Ye, G. Zhong, S. Lu, and L. Zhang, "GRAdient Broadcast: A Robust Data Delivery Protocol for Large Scale Sensor Networks," *Wireless Networks*, vol. 11, no. 3, pp. 285–298, 2005.

[11] Prowler: Probabilistic Wireless Network Simulator. [Online]. Available: http://www.isis.vanderbilt.edu/Projects/nest/prowler/