Energy Efficient Target Coverage in Partially Deployed Software Defined Wireless Sensor Network

Slavica Tomovic(✉) and Igor Radusinovic

University of Montenegro, Džordža Vašingtona bb, 81000 Podgorica, Montenegro
{slavicat, igorr}@ac.me

Abstract. Limited energy resources of sensor nodes are one of the main weaknesses of wireless sensor networks (WSNs). It has long been recognized that conventional methods of data transmission in WSNs are energy inefficient. However, implementation of coordinated, energy-aware routing and power control strategies among sensor nodes is difficult due to distributed network control. Software defined networking (SDN) is a new networking paradigm which overcomes this issue by decoupling the network control and data planes. As an emerging technology, originally envisioned for wired networks, SDN cannot be expected to completely replace traditional WSNs in near future. Therefore, in this paper, we investigate how to save energy in partially deployed software-defined WSN (SD-WSN). In particular, the paper considers the scenario of WSN deployed for monitoring set of targets with known locations, and analyses how the incremental SDN deployment and various power- mode switching policies could affect the WSN lifetime.

Keywords: SDN · WSN · Target coverage · Routing · Energy efficiency

1 Introduction

Wireless sensor networks (WSNs) consist of small, usually low-powered devices (sensor nodes), that measure specific parameters of the environment (e.g. temperature, pressure, motion, etc.). Each sensor node (SN) has the wireless communication capability, which enables it to send report messages to the network gateway. The gateway further delivers the gathered information to more powerful Internet-connected device that can process it. In this way, WSN can significantly reduce, or even completely eliminate the need for human involvement in many civilian, industrial, agricultural, and military applications [1].

Energy is the main resource constraint of SNs because their power sources are usually batteries with limited capacity. WSN lifetime is one of the key factors that determines the functionality and the accuracy of the sensing applications. Thus, methods that optimize the energy utilization of SNs are of great importance.

This paper focuses on the *target coverage problem* [2] in WSNs with large number of SNs randomly deployed to monitor (cover) set of targets with known locations. The main application requirement is to cover all the targets and regularly deliver sensed data to the gateway as long as possible. One way to increase the lifetime of such a WSN is to schedule sensor nodes' activities. Since all SNs in the network share the common sensing task, if a target is monitored by multiple SNs, turning off some nodes does not affect the overall system function as long as target coverage is guaranteed. Although scheduling of sensor nodes's activities has been studied in literature, most of the proposed solutions are focused on minimizing the number of active SNs used for target coverage [2–7]. They assume that energy is consumed only for sensing, i.e. that each SN consumes a same amount of energy in the active power mode. However, in practice, energy consumed for data transmission predominantly determines the WSN lifetime. Energy is not only consumed by a SN which generates data, but also by all SNs along its route to the gateway. Thus, optimal scheduling decisions cannot be made without global view of the network state, which is lacking in traditional WSNs with a distributed control plane. Also, it should be noted that efficient energy utilization is usually of more importance than overall energy consumption, because unbalanced energy utilization can cause network partition even when a large number of SNs have a maximum residual energy [7]. Thus, small energy consumption does not always lead to increased WSN lifetime. Distributed routing protocols are one of the main causes of unbalanced energy utilization. For example, conventional multi-hop communication scheme based on Minimum Transmission Energy (MTE) routing, often leads to equally short WSN lifetime as direct communication with the gateway [8]. In MTE network, all nodes serve as routers for other nodes. Because SNs close to the gateway are most engaged in transmission of data, they have a tendency to drain their energy resources soon compared to the more distant nodes. This results in network partition and degradation of the network coverage.

Considering that distributed control plane of traditional WSNs prevents global resource optimization and smart traffic management, we propose the use of software defined networking (SDN). SDN is technology initially proposed for wired networks, which separates control logic from the forwarding hardware [9]. In SDN networks, the control plane is placed on a logically centralized controller, which maintains a global view of the network, interacts with simple forwarding devices and provides a programming interface for network management applications. Leveraging centralized intelligence of the SDN controller, it is possible to dynamically alter the network behaviour and deploy new applications in real time [10, 11].

The target coverage problem that we consider is motivated by the scenario where SDN sensor nodes (SDNSNs) are incrementally deployed in traditional ad-hoc WSN. Particularly, we have focused on MTE-based WSNs, with only a small percentage of SDNSNs deployed. The key questions we are trying to answer are:

1. Whether it is possible to do effective traffic engineering and prolong the network lifetime with only a small percentage of SDNSNs?
2. Which features SDNSNs and regular sensor nodes must have in order to be able to cooperate in the same WSN?
3. How various activity scheduling algorithms affect the lifetime of the considered WSN model?

The rest of the paper is organized as follows. In Sect. 2 we outline the network model assumed in the analysis and the proposed routing scheme. The analysed activity scheduling algorithms are described in Sect. 3. Simulation results are presented in Sect. 4. Concluding remarks are given in Sect. 5.

2 System Architecture

We consider WSN (Fig. 1) comprising of three main components: regular SNs, SDNSNs and SDN controller, which is integrated at externally supplied gateway (e.g. base station) and makes routing decisions for SDNSNs.

SNs are randomly deployed close to the targets with known locations that need to be continuously observed. At regular intervals, they send report messages regarding the

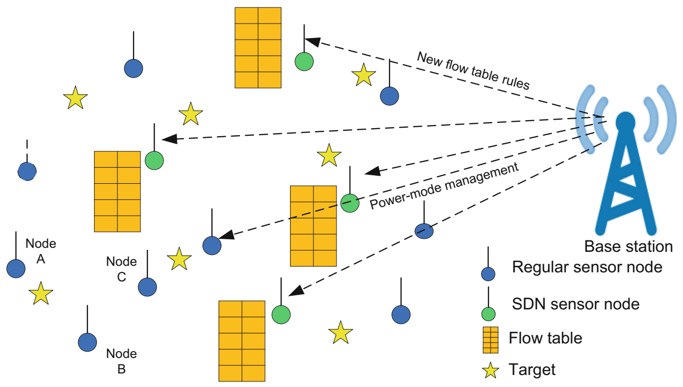

Fig. 1. The proposed network model.

observed status of the targets to the base station (BS). SNs can be in active or low-energy sleep mode. The network activity is organized into rounds, which consists of configuration and sensing phase. In the configuration phase, SDN controller makes routing decisions and decisions regarding the power mode of SNs in the sensing phase. The set of active sensor nodes in each round should be chosen by taking into account two constraints: (i) each target must covered by at least one node at any time; (ii) there must be a feasible route from each SN that participates in target monitoring towards the gateway. Thus, the set of active nodes includes nodes which are supposed to monitor the targets and generate report messages, and nodes which are needed only for relaying the messages towards the BS. The routing and scheduling algorithms are centralized and designed with the objective of maximizing the WSN lifetime, which is measured by the time that elapses from the network initialization to the moment when one or both of the mentioned constraints could not be met due to lack of energy.

Our initial assumption is that no changes are made to regular SNs, i.e. they are completely unaware of the existence of SDN elements in the network. The SDN controller is able to control regular SNs only by sending configuration commands that alternate the power mode. It is not able to control their forwarding behaviour. We have assumed that regular SNs run MTE routing protocol, which is a conventional protocol in today's WSNs. The routing principle of MTE protocol is to choose the next-hop on the route such that energy consumed for transmission is minimized. For example, node A in Fig. 1 would transmit to node C through node B if and only if:

$$d_{AB}^2 + d_{BC}^2 \leq d_{AC}^2 \tag{1}$$

In the above formula, *d* denotes distance between the specified points. To make such forwarding decision, MTE node needs to know all functional SNs within its coverage area and their position. These information may be obtained via periodic exchange of "keep-alive" messages between SNs. Since SDNSNs have to participate in this exchange, we assumed that MTE routing daemon is running on them as well. Note that MTE routing logic makes sense only if nodes can vary the amount of transmit power. Technological advances in radio hardware make this assumption reasonable. The power level at which data will be transmitted is determined based on the location of the next-hop neighbour. For regular nodes that is always the neighbour on the way to the BS which requires the minimum power consumption, while for SDNSNs that could be any node within their radio range.

SDNSNs have a role similar to OpenFlow switches [12] in wired SDN networks. They perform data forwarding according to the controller's instructions stored in flow tables. Due to the specificity of WSN environment, format of flow table is modified compared to the OpenFlow specification. We assumed the table format proposed in [13]. Each flow table entry consists of: (i) matching rule - which defines the characteristics of packets that belong to the same traffic flow; (ii) action - which defines the processing steps; and (iii) counters - which serve for statistical purposes. As shown in Table 1, matching rules contain several window blocks which refer to blocks of bytes that will be matched against the packets. Each *window* consists of four fields, which define: the number and location of bytes that are supposed to be analysed, relational operator that is used during the block analysis, and value which is matched against the specified block of bytes. For our study, only two *actions* are of interest: forwarding and deactivation of the radio interface. Thus, the *action value* field indicates either the next hop on the route or time interval during which the radio interface should be turned off. For example, the first entry from Table 1 specifies that all packets that have in bytes 2 and 3 values 172 and 24 must be forwarded to node 170.16. The value in the last column indicates that 17 packets of this flow have been processed by SDNSN up to now. SDNSNs perform MTE routing only when no route to SDN controller is known. This could happen when control communication is lost due to bad conditions on the wireless channel.

Table 1. Format of flow table proposed in [13].

Window 1				…	Action		Stats
Size	Operator	Address	Value		Type	Value	
2	=	2	172.24	…	Forward	170.21	17
2	=	2	170.16	…	Drop	1	3
2	≠	2	170.25	…	Forward	170.22	3

SDN control communication relies on three types of messages: beacon packets, packet-in requests and flow-mod responses [13]. Beacon packets are periodically broadcasted by the BS. These packets contain one-byte field which indicates the number of hops required to reach the BS/controller from the transmitting SN. BS initially sets this byte to zero. Upon receiving a beacon packet, each SN checks whether the incremented byte value is less than the current estimate of the distance to the BS (initially ∞). If yes, the current estimate and estimate in the beacon packet are updated. Only if the received beacon packet is updated, it will be broadcasted further. In this way, SDNSNs learn a path towards the controller. SDNSN sends packet-in message to SDN controller when a received data packet does not match any of the rules in the flow table. This message carries the packet's header, based on which the controller can reactively make routing decisions. To inform SDN nodes on routing decisions, the controller generates flow-mod messages, which contain elements of flow table entries.

In order to be able to make smart routing and activity management decisions, SDN controller needs information about the network topology and SNs' characteristics such as SDN capability, location, radio range and energy consumption model. As will be discussed in the next chapters, node's energy consumption mostly depends on radio characteristics, distance to the next-hop node and amount of data transmitted. We assume that the first factor is known by controller in advance. The second factor is calculated based on the locations of the node and its next-hop neighbour. Information about the number of bits transmitted by SN during some period of time may be derived from the flow table counters. Because regular nodes do not have flow tables, we assume that BS has a flow table with separate entry for each SN. These entries perform matching on the packet source address, such that the counters indicate number of bytes originated at each node which successfully reached the BS. SNs also consume energy when forward data generated by other nodes in the network. However, SDN controller knows routes from each node to the BS that were used during the analysed time interval. Therefore, by knowing the routes and the flow table counters it can estimate a total amount of traffic carried by each SN. Computation of residual energy is performed at the beginning of each configuration phase by considering the last estimation, time when the last estimation was made, route collection, current and previous state of the corresponding counter in the flow table of the BS. Although SDN controller may not be able to predict precisely when SN runs out of energy (e.g. it is possible that some transmitted data are lost), estimations of residual energy could help in making efficient control decisions.

We have proposed the routing algorithm for the considered heterogenous WSN model in [11]. The key step of the algorithm is to create the reduced network graph. The reduced network graph differs from the connectivity graph by the number of links. It includes all links that originate at SDNSNs, and links that connect MTE nodes with their next-hop neighbours. The link cost function is defined as:

$$Cost(l) = \frac{EC(l)^{\alpha}}{RE(l_{src})^{\beta}} \tag{2}$$

In the above formula *EC(l)* denotes energy needed to transmit and receive a packet on the link *l*, while $RE(l_{src})$ denotes the residual energy of the link source node. Parameters α

and β define the relative impact of these two factors on total link cost. They can be chosen in way to prefer the minimum energy paths or paths with nodes having the most energy, or a combination of the above. Once the link costs are determined, the least cost paths are calculated by the Dijkstra algorithm [14].

3 Node Activity Scheduling

In this section we present three algorithms for WSN activity management, which have been used in our simulation analysis.

Energy-Aware CPNS algorithm (EACPNS) is centralized version of the Coverage-Preserving Node Schedule (CPNS) algorithm [5]. CPNS algorithm aims to minimize energy consumption by minimizing the number of SNs that are used for target monitoring during each round of the WSN operation. It also provides guarantees that its decisions will not jeopardize the target coverage. The algorithm is distributed, and runs on each SN during the configuration phase. A node uses local neighbour information to decide whether to turn off itself or not. If the whole sensing area (in our case just targets within the sensing area) is covered by neighbouring nodes, this node can be turned off without reducing the network coverage [5]. In order to avoid "blind points", which may occur when two neighbouring nodes make decisions in the same time (because each believes that the other one will be in active state during the sensing phase), each node in the network delays its decision for a small, random period of time, and then notifies the neighbours. EACPNS is centralized, slightly modified version of CPNS algorithm. This version of the algorithm firstly sorts SNs in increasing order in terms of residual energy, and then determines the activity status of each node one by one. In this way, low-energy nodes have priority to be deactivated first if their targets could be monitored by neighbouring nodes with more energy. We have constrained selection of active nodes only on those that are able to reach the BS, either directly or by multi-hop communication.

Minimum Energy Consumption Algorithm (MECA) aims to minimize overall energy consumption during the sensing phase. To achieve this, besides energy consumption of SNs that are used for target monitoring, it takes into account energy consumption of relay nodes as well. Since energy consumption depends on several factors, such as: energy needed for communication with the BS and amount of generated data, we have used simulated annealing algorithm [15] to find an acceptable solution. The input argument of the algorithm are the coordinates of nodes that have targets within their sensing areas. More precisely, for each target, a set of candidate nodes is determined. As was the case with EACPNS algorithm, only nodes that have sufficient energy and which can communicate with the BS are taken into account. Also, in order to assure balanced energy consumption among the "monitoring nodes", at the beginning of the configuration phase SDN controller computes average energy level in the network, and removes nodes whose energy level is below the average value from the list of the candidates. The output of the algorithm is a subset of candidate nodes that is considered the most optimal from all solutions analysed during the algorithm's runtime. Optimality of the solution is evaluated based on total energy consumption that the solution requires for one round of WSN operation. The initial solution is chosen randomly (one candidate node for each target), and used as an input argument of the algorithm. At each iteration, the next state, which is in our case a new subset of the candidate nodes, is derived from previous one by perturbing coordinates of the corresponding nodes. The candidate nodes that have locations closest to the newly obtained coordinates become the new subset of candidate nodes that is going to be analysed. Given the current state at iteration *k*, represented by subset of candidate nodes *C* with energy cost *Ec*, the new state, represented by subset of candidate nodes *C'* with energy cost *Ec'*, will become the current state with the probability:

$$p_k = \begin{cases} e^{-(Ec' - Ec)/\alpha_k}, & Ec' > Ec \\ 1 & , Ec' < Ec \end{cases} \tag{3}$$

where α_k is control parameter which increases with increasing *k*. We have configured this parameter according to the suggestions given in [16].

The third algorithm that we have analysed is based on CWGC algorithm [17]. CWGC algorithm is organized in three phases. In phase 1, a “communications tree” is constructed, which connects all SNs with the BS over the path with the least weight. We have adjusted this phase to the routing algorithm presented in Sect. 2. In phase 2, CWGC uses a greedy algorithm to determine a set of SNs that will monitor the targets and generate data in the following round of the WSN operation. This is done iteratively, by choosing node with the largest profit in each iteration. The node’s profit is calculated as the ratio of the number of uncovered targets in the sensing range of the node, and weight of the path which connects the node with the BS in the communication tree. After selecting a monitoring node, the path weights of the upstream nodes in the communication tree are updated according to formula:

$$w(p_s) = w(p_s) + (e_{TX} + e_{RX}) \cdot B_r \cdot w(p_s)/E_r(s) \tag{4}$$

where B_r is the number of bits that the selected node is expected to generate in the round *r*, e_{TX} and e_{RX} are energies consumed by upstream node *s* for receiving and transmitting bit to the next-hop neighbour, and $E_r(s)$ is residual energy of the node *s*. Note that if a path weight of a SN changes, its profit value changes as well. At the end of each iteration, the selected SN is marked as active node and all targets within its sensing area are considered already covered in the following iterations. The process repeats until all targets are covered. In the final phase, the algorithm outputs a sub-tree of the *communication tree* on the basis of the selected monitoring nodes.

4 Performance Evaluation

In order to verify the effectiveness of the partially deployed SD-WSN and different algorithms for node activity scheduling in such an environment, we carried out set of simulations in MATLAB. Throughout the simulations, we have considered several random network configurations distributed in an 200 m × 200 m area, where each node is assigned an initial energy of 0.25 J. In the considered network scenarios, 20 target points are uniformly distributed over the area, while BS is positioned in the centre of the area. All SNs are assumed to have the same sensing range *Rs = 40* m and the same maximum communication range *Rc = 80* m. Values of α and β parameters in link cost definition have been set to 1 and 4 respectively.

WSN operation is organized into rounds. At the beginning of each round, SDN controller makes decisions regarding the activity status of SNs. We assumed that SNs send data at fixed rate. Each active SN which has one or more targets in its sensing area, generates ten 2000b data packets in regular intervals during the sensing phase of the round. Energy consumption in sleep mode has been neglected. By default, the round duration is set to be long enough to embrace ten regular reportings of the monitoring nodes. However, the SDN controller is able to define a smaller round duration when a selected set of active nodes does not have sufficient energy.

In simulations, we assumed the same radio model discussed in [8, 11, 16]. Energy consumed for the transfer of a k bit message between two SNs separated by a distance of r meters has been calculated as follows:

$$E_T = E_{Tx}k + E_{amp}k = \begin{cases} E_{Tx}k + \varepsilon_{FS}r^2k, & r \leq r_o \\ E_{Tx}k + \varepsilon_{TW}r^4k, & r \geq r_o \end{cases} \tag{5}$$

$$E_R = E_{Rx}k \tag{6}$$

where E_T denotes the total energy dissipated in the transmitter and E_R represents the total energy dissipated in the receiver electronics. Parameters E_{Tx} and E_{Rx} are per-bit energy dissipations for transmission and reception respectively. In transmission, additional energy is dissipated to amplify the signal (E_{amp}). Parameters ε_{FS} and ε_{TW} denote amplifier parameters corresponding to the free-space and two-ray propagation models respectively, while r_o is threshold distance given by the expression:

$$r_o = \sqrt{\varepsilon_{FS}/\varepsilon_{TW}} \tag{7}$$

All the radio model parameters have been configured to values used in [8, 11, 16] ($E_{Tx} = E_{Rx} = 50nJ/bit$, $\varepsilon_{FS} = 10pJ/bit/m^2$, and $\varepsilon_{TW} = 0.0013pJ/b/m^4$).

In the first experiment, we have studied the impact of node density on the lifetime of WSN in scenarios where MTE routing protocol is used by all SNs, and where 25 % of SNs have SDN capability and perform routing according to the centralized algorithm described in [11]. In the first scenario (MTE-CPNS), CPNS algorithm is used for managing activity of SNs, because we assume that all control decisions in MTE network are made in a distributed manner. The performance of SDN-based WSN is evaluated for each of the activity scheduling algorithms described in Sect. 3 (SDN-MECA, SDN-EACPNS, SDN-CWGS scenarios).

The Fig. 2 shows the WSN lifetime expressed in number of the reporting cycles as a function of the number of SNs in the network. The shown results are an average of multiple runs, each with a random generated topology. It should be noted that depending on the network topology, the obtained results sometimes varied substantially, but performance ordering of the considered network configurations remained the same in all simulations performed. From Fig. 2 we can see that the network lifetime increases as the

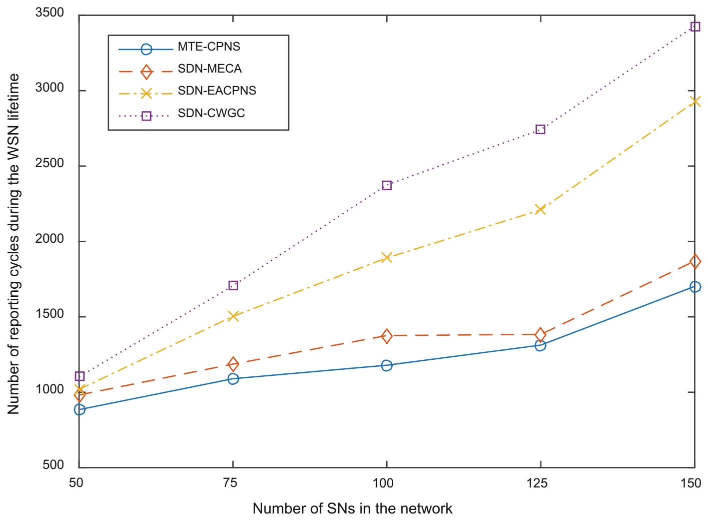

Fig. 2. The WSN lifetime vs. node density.

network density increases. This is because more SNs can be used to sense the targets and forward data to the BS. The results show that the hybrid WSN, consisted of 75 % of regular and 25 % of SDNSNS, for the same initial energy always produces a longer WSN lifetime than traditional WSN with MTE routing scheme and CPNS activity scheduling algorithm. The short WSN lifetime in MTE-CPNS scenario is dominantly a consequence of distributed routing. In traditional MTE WSNs, nodes which are close to the gateway forward the largest amount of data and suffer higher energy losses. These nodes “die” very fast, causing the network partitioning. Thus, even when there are nodes with sufficient energy that could be used for target coverage, it happens that the generated data cannot reach the BS because the upstream relaying nodes are out of energy. If we take WSN with 100 SNs as a representative example, in traditional MTE network the first node “dies” in 270th reporting cycle, in average. On the other side, in all the analysed SDN scenarios the first dead node occurs much later (Fig. 3). This could be attributed to a more balanced energy consumption among SNs.

In simulations of partially deployed SDWSNs, the MECA algorithm for node activity scheduling has shown the worst performance. This suggests that energy consumption is not a good criterion for selecting active nodes. As discussed earlier, when the objective is to maximize the WSN lifetime, overall energy consumption is not that important as to assure that each node consumes energy evenly. MECA algorithm fails to achieve a balanced energy consumption. The results from Fig. 3 support this claim, since we can see that the increase in WSN density does not always prolongs the lifetime of all SNs. However, the main reason of the poor performance of MECA algorithm is a large amount of generated data within the network. Figure 4 shows an average number of active SNs during one round of WSN operation. The smaller this number is, the smaller amount of data is generated. The shown results indicate the advantages of greedy algorithms for node activity scheduling, which in each iteration

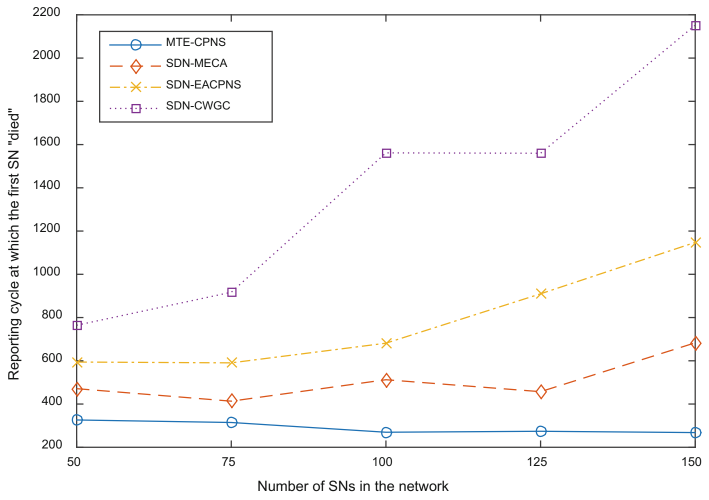

Fig. 3. The reporting cycle at which first SN runs out of energy vs. network density.

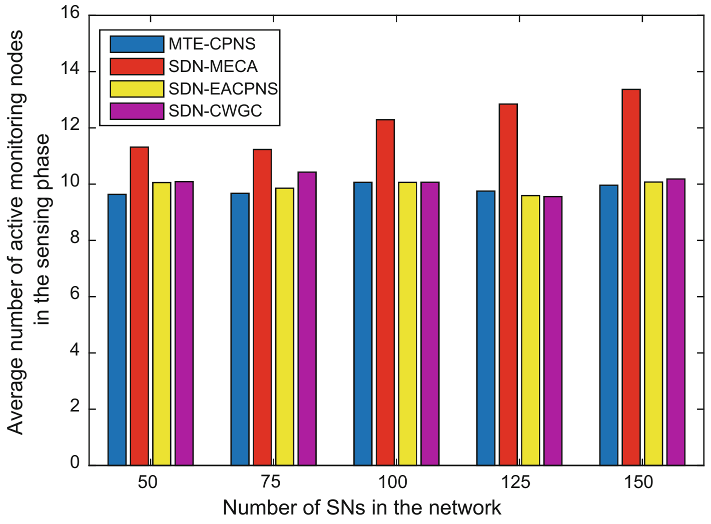

Fig. 4. The average number of active "monitoring" SNs during one round of WSN operation.

choose one active node according to specific criteria (CWGC, EACPNS, CPNS). When a SN is selected to be in active state, all of its targets are considered covered in the following iterations of the greedy algorithms. Thus, there is a lower probability for redundant data to be generated, which happens in cases when multiple SNs monitor the same target. On the other side, MECA algorithm rather examines the optimality of the complete solutions, consisted of one candidate node for each of the targets. Although the MECA's optimization function tends to decrease the number of active nodes, when the running time of the algorithm is limited to a reasonable number of iterations (200 in our simulation setup), the final solution may result in a high level of data redundancy. However, we can see that the introduction of even a small percentage of SDNSNs has been sufficient to compensate the inefficient scheduling decisions and prolong the WSN lifetime. Apparently, CWGC algorithm most efficiently exploits the benefits of centralized control among all the analysed activity scheduling algorithms. This is because CWGC uses the knowledge of the routing paths to select monitoring SNs which will not cause the "transmission bottlenecks" [17]. On the other side, EACPNS algorithm efficiently balances energy consumption among the monitoring nodes, but it neglects the fact that some of them might use the same relay SNs to deliver data to the gateway. Since the routing paths remain valid during the whole sensing phase, energy losses of the "bottleneck" SNs could be significant. In particular, the proposed WSN architecture with CWGC activity scheduling algorithm achieves 22.3 % longer lifetime than with the EACPNS algorithm in topology with 100 SNs.

5 Conclusion

When WSN is densely deployed for the purpose of monitoring a set of targets with known locations, the energy savings could be achieved by scheduling SNs to work alternatively. However, due to distributed control plane, in traditional WSNs it is hard to achieve balanced energy consumption. Routing and scheduling decisions are made by individual SNs, based on incomplete, local view of the network state, which prevents global resource optimization and smart traffic management. In this paper, we have analysed how the mentioned problems could be alleviated with incremental deployment of SDN-enabled SNs. The hybrid WSN model is presented, which assumes cooperation between traditional SNs, that run distributed MTE routing protocol, and SDN-enabled SNs. Through simulations we have shown that if information regarding the positions and capabilities of SNs are available, the proposed WSN model promises a significant increase in the WSN lifetime even when a small percentage of SDN-enabled SNs is deployed. Further on, we have pointed out the importance of a centralized view of the routing paths for activity scheduling algorithms. In particular, CWGC algorithm [17] has been identified as very beneficial for the considered WSN scenario.

The results presented in this paper only indicate potential benefits of incremental SDN deployment into traditional WSNs. In the next phase of our research, we will evaluate performance of the proposed architecture more accurately, by taking into account the influence of proportion of SDNSNs on WSN lifetime as well as energy waste due to control overhead and other more accurate wireless channel models.

Acknowledgments. This work has been supported by the EU FP7 project Fore-Mont (Grant Agreement No. 315970 FP7-REGPOT-CT-2013) and the BIO-ICT Centre of Excellence (Contract No. 01-1001) funded by the Ministry of Science of Montenegro and the HERIC project.

References

1. He, T., et al.: Energy-efficient surveillance system using wireless sensor networks. In: ACM International Conference on Mobile Systems, Applications and Services (2004)
2. Cardei, M., Thai, M.T., Li, Y., Wu, W.: Energy-efficient target coverage in wireless sensor networks. In: IEEE INFOCOM 2005, Miami, pp. 1976–1984 (2005)
3. Cardei, M., Wu, J., Lu, M., Pervaiz, M.O.: Maximum network lifetime in wireless sensor networks with adjustable sensing ranges. In: IEEE International Conference on Wireless and Mobile Computing, vol. 3, pp. 438–445 (2005)
4. Cardei, M., Du, D.-Z.: Improving wireless sensor network lifetime through power aware organization. Wirel. Netw. **11**(3), 333–340 (2005). Kluwer Academic Publishers
5. Tian, D., Georganas, N. D.: A coverage-preserving node scheduling scheme for large wireless sensor networks. In: ACM International Workshop on Wireless Sensor Networks and Applications, pp. 31–41 (2002)
6. Changlin, Y., Kwan-Wu, C.: A novel distributed algorithm for complete targets coverage in energy harvesting wireless sensor networks. In: IEEE International Conference on Communications (ICC), pp. 361–366 (2014)
7. Thippeswamy, B.M., et al.: EDOCR: energy density on-demand cluster routing in wireless sensor networks. Int. J. Comput. Netw. Commun. **6**(1), 223–240 (2014)
8. Heinzelman, W. R., Chandrakasan, A. P., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the 33rd Annual Hawaii International Conference, Hawaii, vol. 2, pp. 10–17 (2000)
9. Open Networking Foundation: Software Defined Networking: the New Norm for Networks. Web White Paper. https://www.opennetworking.org/
10. Gante, A., Aslan, M., Matravy, A.: Smart wireless sensor network management based on software-defined networking. In: 2014 27th Biennial Symposium in Communications (QBSC), Ontario, pp. 71–74 (2014)
11. Tomovic, S., Radusinovic, I.: Performance analysis of a new SDN-based WSN architecture: In: Proceedings of 23rd Telecommunication Forum TELFOR 2015, Belgrade, Serbia, pp. 99–102 (2015)
12. OpenFlow Switch Specification v1.0.0. http://archive.openflow.org/documents/openflow-spec-v1.0.0.pdf
13. Costanzo, S.., Galluccio, L., Morabito, G., Palazzo, S.: Software defined wireless networks: unbridling SDNs. In: European Workshop on Software Defined Networking (EWSDN), Darmstadt, pp. 1–6 (2012)
14. Dijkstra, E.W.: A note on two problems in connection with graphs. Numerische Math **1**, 269–271 (1959)
15. Murata, T., Ishibuchi, H.: Performance evaluation of genetic algorithms for flowshop scheduling problems: In: IEEE Conference on Evolutionary Computation, vol. 2, pp. 812–817 (1994)
16. Heinzelman, W.B., Chandrakasan, A.P., Balakrishnan, H.: An application-specific protocol architecture for wireless microsensor networks. IEEE Trans. Wirel. Commun. **1**(4), 660–700 (2002)
17. Qun, Z., Gurusamy, M.: Maximizing network lifetime for connected target coverage in wireless sensor networks. In: IEEE International Conference on Wireless and Mobile Computing, Montreal, pp. 94–101 (2006)