Extending the lifetime of wireless sensor network with partial SDN deployment

Slavica Tomovic and Igor Radusinovic, *Member, IEEE*

***Abstract* — Energy efficiency is one of the key requirements in Wireless Sensor Networks (WSNs). In order to optimize energy usage at sensor nodes, this paper proposes a new network architecture that relies on concepts of Software Defined Networking (SDN). Since SDN is a relatively new technology, originally envisioned for wired networks, it cannot be expected to get immediately and completely adopted in WSN domain, regardless of potential benefits. For this reason, we consider incremental SDN deployment where SDN nodes coexist with traditional sensor nodes, and propose a new routing algorithm for SDN controller that prolongs the WSN lifetime even when a small percentage of SDN nodes is deployed.**

***Keywords* — energy efficiency, routing, SDN, WSN.**

I. Introduction

WIRELESS Sensor Networks (WSNs) consist of spatially distributed, low power devices (SNs -sensor nodes), equipped with sensors that measure specific parameters of the environment. Depending on application, they may include a vast number of SNs. The sensor data are transmitted to the external gateway either directly, or by multi-hop communication. In an ad-hoc scenario, each SN participates in routing by forwarding data for the other nodes. There is a variety of scenarios in which such networks might find uses, such as agriculture, transportation, industry, healthcare, etc [1].

Energy is an important resource constraint of SNs because their power sources are usually batteries with limited capacity. On the other hand, one of the key factors that determines the functionality and the accuracy of the sensing applications is WSN lifetime, which may be prolonged by balanced energy consumption among sensor nodes [2]. These unique features must be taken into account while designing routing protocols and algorithms. If the gateway is too far from a SN direct transmission has to be avoided because the incurred energy loss can be quite extensive. However, conventional multi-hop communication such as Minimum Transmission Energy (MTE) routing often results in an equally undesirable effect [3]. In a network which uses MTE routing protocol, each node acts as a router for the other nodes. Because nodes close to the gateway are most engaged in transmission of data, their energy resources rapidly drain. Various routing protocols have been proposed to alleviate the mentioned problems [4]. However, most of the proposed solutions are too complex with questionable potential for practical implementation.

Many inherent problems of WSNs are deeply rooted in the network architecture. In ad-hoc networks each SN participates in decentralized routing and independently determines the next-hop for sending the data towards the destination. This prevents global resource optimization and smart traffic management [5]. Thus, we propose use of Software Defined Networking (SDN) in WSNs. SDN is a technology initially proposed for wired networks, with separated control and data planes [6]. In SDN networks, the control plane is placed on a logically centralized controller, which maintains a global view of the network, interacts with simple data-plane and provides a programming interface for user-written network management applications. Leveraging centralized intelligence of the SDN controller, it is possible to dynamically alter the network behaviour and increase the energy efficiency [7]-[9].

The routing problem that we consider is motivated by the scenario where SDN sensor nodes (SDNSNs) are incrementally deployed in a traditional WSN. The key question we are trying to answer is whether it is possible to do effective traffic engineering and prolong the network lifetime if all nodes in the network cannot be controlled by SDN controller. In particular, we have focused on WSNs that use MTE routing protocol, with only a small percentage of SDNSNs deployed. In the paper, we show that when SDN controller knows the positions and capabilities of SNs, WSN lifetime could be significantly increased regardless of the network topology.

The rest of the paper is structured as follows. In Section II we outline the network and radio models assumed in the analysis, and propose a new routing model. Simulation results and analysis are presented in Section III. Conclusion remarks are given in Section IV.

Paper received May 2, 2016; revised June 14, 2016; accepted June 22, 2016. Date of publication July 20, 2016. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Grozdan Petrović.

This paper is a revised and expanded version of the paper presented at the 23rd Telecommunications Forum TELFOR 2015 [16].

This work has been supported by the Ministry of Science of Montenegro under grant 01-451/2012 (FIRMONT) and the HERIC project through the BIO-ICT Centre of Excellence (Contract No. 01-1001).

S. Tomovic is with the Faculty of Electrical Engineering, University of Montenegro, Dzordza Vasingtona bb, 81000 Podgorica, Montenegro (phone: 382-69-468583; e-mail: slavicat@ac.me).

I. Radusinovic is with the Faculty of Electrical Engineering, University of Montenegro, Dzordza Vasingtona bb, 81000 Podgorica, Montenegro (e-mail: igorr@ac.me).

II. Network model

As an emerging technology, originally envisioned for wired networks, SDN cannot be expected to completely replace traditional WSNs in near future. Therefore, we investigate how to save energy in a partially deployed software-defined WSN. In the rest of the Section we present WSN and radio models assumed in our analysis.

A. WSN architecture

We consider the hybrid WSN (Fig. 1) consisted of:

1. regular SNs that run MTE routing protocol;
2. SDNSNs which run MTE routing protocol, but in addition could be controlled by SDN controller.
3. SDN controller which is implemented at an externally supplied gateway (e.g. a base station) and makes routing decisions for SDNSNs.

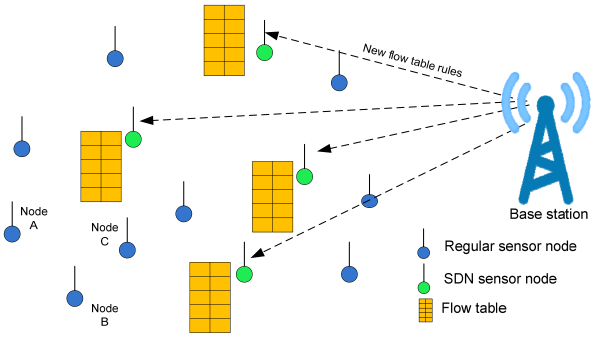


Fig. 1. The proposed WSN model.

Our initial assumption is that no changes are made to regular SNs, i.e. they are completely unaware of the existence of SDN devices in the network. SDN controller is not able to control regular SNs. Instead, 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 [10]. For example, node A in Fig. 1 would transmit to node C through node B if and only if:

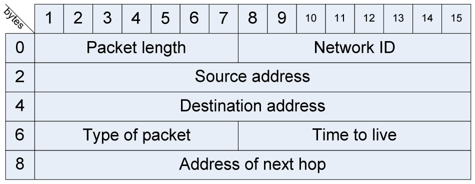
$$d_{AB}^{2} + d_{BC}^{2} \le d_{AC}^{2} \tag{1}$$

In the above formula, d denotes distance between the specified points. To make such a forwarding decision, MTE node needs to know all functional SNs within its coverage area and their positions. These information are obtained via periodic exchange of "keep-alive" messages between the SNs. SDNSNs have to participate in this exchange, so we assumed that MTE routing daemon is running on them as well. Note that MTE routing logic makes sense only if SNs can use power control to vary the amount of transmit power. Technological advances in radio hardware make this assumption reasonable [10]. The power level for data transmission is determined based on the location of the next-hop neighbor. For regular SNs that is always the neighbor that requires the minimum communication energy, while for a SDN node that could be any node within the radio range. Of course, to avoid loops in distributed routing, the next-hop on the route must be less distant from the base station. Note that the real location of neighbors may not be of strong importance for the regular SNs because relative distance to a neighbor could be estimated based on *received signal strength* of the "keep-alive" messages.

The SDNSNs have a role similar to OpenFlow switches [11] in wired SDN networks. They perform data forwarding according to the controller's instructions stored in so called flow tables. Each flow table entry is defined by: i) a matching rule which describes the characteristics of packets belonging to a traffic flow; ii) an action which defines the way of processing a flow; and iii) counters which serve for statistical purposes. Due to specificity of WSN environment, the format of flow table is modified compared to the OpenFlow specification. We assumed the table format proposed in [12]. 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 analyzed (*size* and *addr* fields), relational operator that is used during the block analysis (=,≠,<,>), and a value which is matched against the specified block of bytes. Since we are only interested in routing functionality, in our study the processing action is always forwarding to the next-hop SN. Thus, the *action value* field in our case indicates the next hop on the route. 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 the node 170.21. The value in the last column indicates that 17 packets of this flow have been processed by the SDNSN up to now.

Table 1: Format of flow table proposed in [12].

Window 1				...	Action		Stats
Size	Op.	Addr	Value		Type	Value	Counts
2	=	2	172.24	...	Forward	170.21	17
2	=	2	170.16	...	Drop	1	3
2	≠	2	170.25	...	Forward	170.22	3



bytes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	Packet length							Network ID							
2	Source address														
4	Destination address														
6	Type of packet							Time to live							
8	Address of next hop														

Fig. 2. Packet header format [12].

SDN-related control communication relies on three types of messages: beacon packets, *packet-in* requests and *flow-mod* responses. Each of them uses a header illustrated in Fig. 2. The base station periodically broadcasts beacon packets. Besides the header fields, these packets contain an additional byte indicating the number of hops required to reach the base station from the transmitting SN. Base station 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 base station (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. Packet-in message is sent by SDNSN to SDN controller when a received data packet does not match any of the rules in flow table. It contains header of the data packet, based on which the controller makes routing decisions. To inform SDNSNs on routing decisions, controller generates flow-mod messages, which contain elements of flow entries. For the adopted format of flow table, payload of these messages doesn't exceed 4B if matching is done on one block of packet header bytes [12].

An important assumption made in our study is that SDN controller has knowledge of SNs' positions and capabilities. This is important for two reasons. First of all, to perform traffic engineering the controller needs information about the network topology and available energy resources at each node. Then, to calculate feasible routes the controller must know which nodes could be manipulated and which are the regular ones. By knowing the position of each SN and characteristics such as SDN capability, radio range, energy consumption model and initial energy, the controller can estimate the residual energy of nodes and make efficient routing decisions. Node's energy consumption depends on radio characteristics, distance to the next-hop node and the amount of data transmitted. We assume that the controller knows the first factor in advance. The second factor is calculated based on locations of the node and its next-hop neighbor. Information on the number of bits transmitted by SN during some period of time may be derived from flow table counters. Because regular nodes do not have flow tables, we assumed that the base station has flow table with a separate entry for each SN. These entries perform matching on the packet source address, so the counters indicate the number of bytes originated at each SN which successfully reached the base station. Besides the data generated by them, SNs participate in forwarding of data generated by other nodes in the network. However, SDN controller knows the routes from each SN to the base station that were used during the analyzed time interval. Therefore, by knowing the routes and the flow table counters it can estimate the total amount of traffic carried by SN. When SDN controller computes the estimate of residual energy for some SN, it considers the last estimation, time when the last estimation was computed, the route collection, the current and previous state of the corresponding counter in the flow table. Since forwarding rules at SDNSNs could be dynamically changed, the computation of residual energy must be triggered each time before a new route is installed in order to keep a correct insight into energy consumption. Although SDN controller may not be able to predict when SN is "dead" (e.g. it is possible that some transmitted data are lost), estimations of residual energy help in finding the most efficient route.

Algorithm 1 describes the routing logic of SDN controller. In the first step the controller creates a reduced network graph, which contains only links that could be used for the route setup. These are all links which originate at SDNSNs (lines 8-9), and links from each regular SN to the next-hop neighbor according to MTE protocol (lines 13-14). Link cost is chosen as a function of residual energy of the source SN and energy needed to transmit and receive a packet on the link. Parameters α and β define the relative impact of these two factors on total link cost. They can be chosen to find the minimum energy path or the path with nodes having the most energy, or a combination of the above. Once the link costs are determined, Dijkstra algorithm [13] is used to find the least cost route between the source SN and the base station.

***Algorithm 1**: Route calculation*

```
1:# G- reduced network graph used for route calculation
2:# src- source node, BS- base station, nodes- live nodes
3:# Costs - link costs, R - residual energy of source node
4:# EC - energy used to transmit and recevice data on the link
5: function:  find_route(src, BS, nodes)
6:   for node in nodes do:
7:    if node.type is SDN do:
8:     for nb in node.neigbours do:
9:       G.addLink(node, nb)
10:      Costs([node, nb]) = EC([node, nb])$^{\alpha}$/ R([node])$^{\beta}$
11:     end for
12   else do:
13:    nb=node.MTE_nextHop
14:    G.addLink(node, nb)
15:    Costs([node,nb])=EC([node,nb])$^{\alpha}$/R([node])$^{\beta}$
16:   end if
17: end for
18: return leastCostPath(G, src, BS, Costs)
19: end function
```

B. The radio model

In our analysis we used the same energy consumption model discussed in [3], [10], [14], where the transmitter dissipates energy to run the radio electronics and the power amplifier, and receiver only dissipates energy to run the radio electronics. Energy dissipation in other components of the typical SN has been neglected.

Energy consumed for the transfer of a k bit message between two SNs separated by a distance of r meters is given by the following equations:

$$E_T = E_{Tx}k + E_{amp}k \tag{2}$$

$$E_R = E_{Rx}k \tag{3}$$

where E_T denotes the total energy dissipated in the transmitter of the source node 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}) according to the distance to the destination. As is the case in [3], we assumed that propagation loss is inversely proportional to r^2 (free-space model) for small distances, while it is inversely proportional to r^4 for long distances (two-ray model). Thus, the overall energy consumed by the radio to transmit a k-bit message over distance r is calculated as follows:

$$E_T = E_{Tx}k + E_{amp}k = \begin{cases} E_{Tx}k + \varepsilon_{FS}r^2k, \; r \le r_o \\ E_{Tx}k + \varepsilon_{TW}r^4k, \; r \ge r_o \end{cases} \tag{4}$$

where ε_{FS} and ε_{TW} are amplifier parameters for free-space and two-ray propagation models respectively, and r_o is threshold distance given by:

$$r_o = \sqrt{\varepsilon_{FS} / \varepsilon_{TW}} \quad (5)$$

In simulations described in the paper, we assumed the same set of parameters as in [10]: $E_{Tx} = E_{Rx} = 50nJ/bit$, $\varepsilon_{FS} = 10\,pJ/bit/m^2$, and $\varepsilon_{TW} = 0.0013\,pJ/b/m^4$.

III. Simulation Results

In order to verify the effectiveness of the proposed solution, we carried out simulations in MATLAB. Performance was measured by the following metrics: energy dissipation, WSN lifetime, total number of successfully delivered messages and number of nodes that are alive.

A. The random network scenario

In the first set of simulations, we used a 100-node network, with nodes randomly distributed in an 200m x 200m area. Location of the base station was chosen to be in the middle of the area (x=100, y=100). Each SN was assigned an initial energy of 1J. Following the approach from the literature [3], [10], [14], we assumed that all SNs are sensing the environment at a fixed rate, and send a 2000-bit data packet to the base station during each time step or "round" of the simulation. The values of α and β parameters in the link cost definition were set to 1 and 4 respectively. The influence of random factors such as wireless channel interference was ignored. In simulations of the partially deployed SDN network the routes are installed proactively in the beginning of each round. Thus, we have taken into account energy consumption due to transmission of data messages and control flow-mod messages. Impact of keep-alive and other control messages has been neglected because it should be the same in SDN and MTE scenario.

The first set of results provides a comparison of the traditional WSN model using MTE routing, and the hybrid WSN with only 20% of SDN nodes. From Fig. 3 we can see the number of SNs alive as a function of the number of sensing rounds. The presented results are an average of 10 simulation runs, each with a random generated topology. The results show that the hybrid WSN, for the same initial energy, produces a longer WSN lifetime. In MTE scenario, the first dead node occurs after only 105 rounds, and after 2000 rounds 76% of the nodes are dead. This is because SNs close to the base station forward the largest amount of data and suffer higher energy losses. These nodes die very fast, causing the increase in energy required by other nodes to reach the base station (Fig. 4). On the other hand, in the hybrid WSN the first node died in the 360th round. If we define WSN lifetime as the number of rounds for which 70% of the SNs remain alive, the proposed solution exceeds the lifetime of MTE network by 49.25%. The standard variation of the difference in WSN lifetime in MTE and SDN scenarios was about 40 rounds, but in each simulation the performance improvement introduced with SDN nodes has been evident. In the hybrid network, the number of SNs alive diminishes more slowly because forwarding rules of SDNSNs are dynamically defined by considering estimated values of the remaining energy for each SN. The outcome is more balanced energy consumption among SNs.

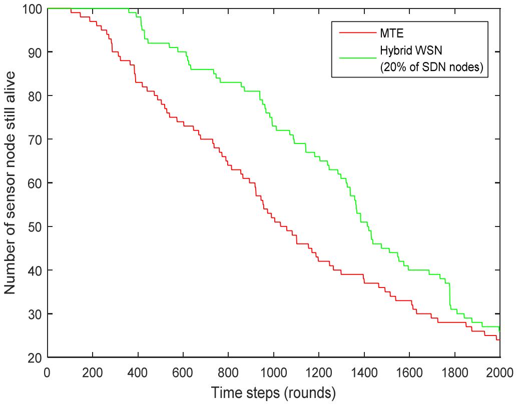


Fig. 3. Lifetime of the hybrid and traditional WSN.

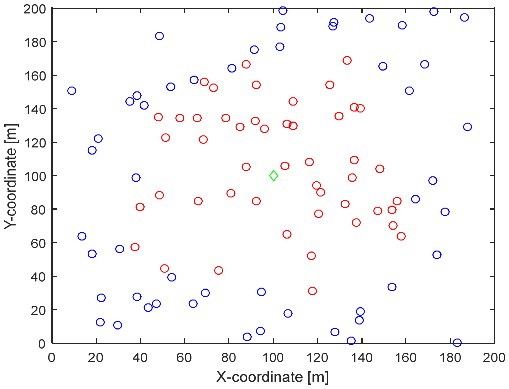


Fig.. 4. Sensor nodes that remained alive (blue circles) and those that are dead (red circles) after 1000 rounds in MTE network. The diamond shape denotes the gateway.

Next we compared the quality of service achieved in the two scenarios described above. Although quality of service is application-specific, the number of data messages received by the base station is often considered as an important quality indicator [10]. The more data the base station receives, the area of the environment is better monitored. Fig. 5 shows the total number of data messages received by the base station in the hybrid and MTE WSN over the rounds of the network activity. We can see that the hybrid WSN has a great advantage in data delivery. With only 20% of SDN nodes deployed, it offers improvement by a factor of 21.86%. In MTE scenario some nodes drained their batteries quickly, and could not longer transmit or receive the data. Throughout simulations we assumed that each SN is within communication range of each other and the base station. For the shorter radio ranges of SNs, it happens that the nodes in MTE WSN unnecessarily consume energy to transmit and receive data which cannot reach the base station even under ideal wireless channel conditions. For example, when lots of nodes around base station die, some parts of the network stay disconnected. As MTE routing does not include any centralized control, SNs at the network edge may be completely unaware of this situation.

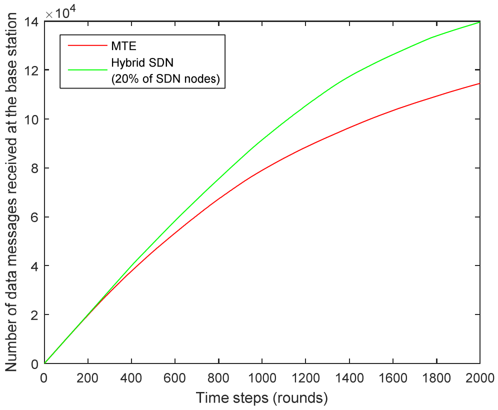


Fig. 5. Total number of data messages received at base station over rounds of operation.

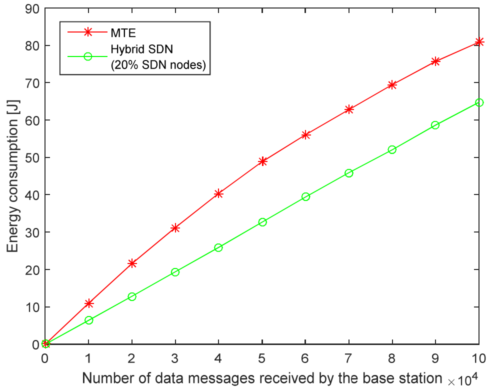


Fig. 6. Dissipated energy as a function of the number of data messages received at base station.

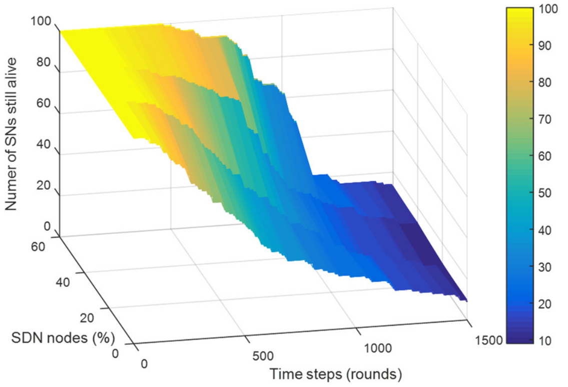
The results from Fig. 6 show that the hybrid WSN has a more desirable energy consumption curve. One of the reasons why MTE requires so much energy to transmit the data is that each message traverses a large number of hops on the way to the base station. On the other side, SDN controller has knowledge of the locations and energy resources of all SNs in the network, so it can select routes in more efficient manner.

Finally, we have analyzed the impact of the number of SDNSNs on the system performance. The obtained results in terms of the number of SNs alive over the rounds of simulation are illustrated in Fig. 7. In a set of experiments the percentage of SDN nodes was increased from 0 to 60 by a step of 20. The initial case, when the number of SDN nodes is zero corresponds to MTE routing. We can see that all subsequent experiments resulted in a significantly longer network lifetime. As expected, the WSN lifetime sharply increases with the number of SDN nodes.

B. *Impact of WSN topology and communication range*

In this section we explore the impact of the network topology on the network lifetime. Since a variety of topologies for WSNs has been developed and each topology has its own scenario, it is hard to make a comprehensive study. In order to validate the performance improvement that the proposed solution achieves, we have compared the performance of MTE and the hybrid WSN in the following scenarios: i) randomly deployed SNs with the gateway (base station) placed in the middle of 200m x 200m area; ii) randomly deployed SNs with the base station located at least 20m from the nearest node (x=100, y=220); iii) uniformly distributed SNs in 200m x 200m area (grid topology) with the base station in the middle; iv) uniformly distributed SNs with the base station located at least 20m from the nearest node. In simulations, the initial energy of SNs has been set to 0.5J.

The obtained results are shown in Fig. 8. We can see that the hybrid WSN model achieves a performance improvement almost in each of the analyzed scenarios. The benefits of introducing SDNSNs in the network are more expressed in the scenarios where the gateway is located in the middle of the sensing area. Distancing the gateway from the middle of the area has resulted in the reduced WSN lifetime. This is expected, since MTE SNs in this case need to transmit data over a large number of relay nodes in order to reach the gateway. Also, we can notice that uniform distribution of SNs may significantly increase the WSN lifetime when the gateway is centrally positioned. If the positions of SDNSNs are chosen to cover the area uniformly, the WSN lifetime could be further increased.



% of SDNSNs	0	20	40	60
WSN lifetime (in time steps)	362	504	718	903

Fig. 7. "Lifetime" of the proposed WSN model as a function of number of SDNSNs in the network. The initial energy of SN has been set to 0.5J.

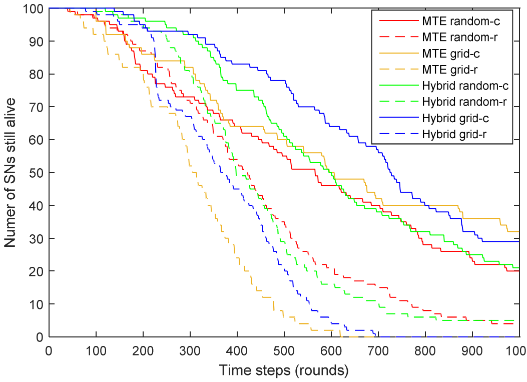


Fig. 8. Lifetime of traditional MTE WSN and the hybrid (SDN-based) WSN for different topologies.

The previously presented simulation results stand for the case where communication range of each SN covers the whole network area. However, in practice, communication range of SNs is usually more limited. Fig. 9 shows how the communication range affects the WSN lifetime. Simulations have been performed on grid topology for three different configurations of the communication range (CR): 50m, 100m and 150m. In the first scenario (CR=50), both network models have become inoperable after a little more than 200 reporting rounds. After this period WSN gets partitioned and none of the SNs was able to transmit data to the gateway. This explains the stop of any changes in the number of SNs alive in Fig 9. However, from the results we can conclude that adding a relatively small number of SDNSNs promises improved network performance regardless of the radio technology used. Namely, even when there is no increase in WSN lifetime, the hybrid WSN obviously achieves better quality of service (i.e. delivers more information) during the WSN lifetime.

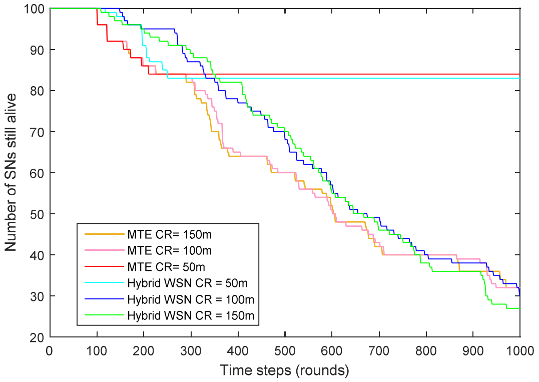


Fig. 9. Lifetime of traditional MTE WSN and the hybrid (SDN-based) WSN for different communication ranges (CR) of SNs (grid topology with gateway in the middle).

IV. Conclusion

In this paper we have proposed a routing algorithm for the hybrid WSN model where SDNSNs coexist with traditional SNs which use MTE routing protocol. The algorithm is based on the assumption that SDN controller knows the locations and main features of SNs. Through simulations we have shown that if this information is available, deployment of even a small percentage of SDNSNs could significantly increase the WSN lifetime and improve the overall network performance. In the next phase of our research we will pay more attention to the selection of routing metric, as it has a deep impact on the system performance. We also plan to implement the proposed WSN architecture on OpenMote platform [15] and verify the obtained results experimentally. Experimental evaluation is necessary because implementation of SDN functionality increases the complexity of sensor node, thus one of the issues that has been neglected in our research so far is the impact of additional complexity on the energy consumption model of SDN-enabled sensor node.

References

[1] Th. Arampatzis, J. Lygeros, S. Manesis, S., "A Survey of Applications of Wireless Sensors and Wireless Sensor Networks," IEEE Mediterrean conference on control and automation, pp. 719-724, 2005.

[2] I. Dietrich and F. Dessler, "On the lifetime of wireless sensor networks," *ACM Transactions on sensor networks*, vol. 5, no. 1, pp. 1-38, 2009.

[3] S. D. Muruganathan, D. C. F. Ma, R. I. Bhasin and A. Fapojuwo, "A centralized energy-efficient routing protocol for wireless sensor networks," *IEEE Comm. Magazine*, vol.43, no.3, pp. 8-13, 2005.

[4] N. A. Pantazis, S. A. Nikolidakis, and D. D. Vergados, "Energy-efficient routing protocols in wireless sensor networks: a survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp.551-591, 2013.

[5] T. Luo, H.-P. Tan, and T. Quek, "Sensor OpenFlow: Enabling software-defined wireless sensor networks," *IEEE Communications letters*, vol. 16, no. 11, 2012.

[6] Open Networking Foundation, "Software Defined Networking: the new norm for networks," Web. White Paper, Retrieved Sept. 2015.

[7] A. Gante, M. Aslan, and A. Matravy, "Smart Wireless Sensor Network Management Based on Software-Defined Networking," 27th QBSC, pp. 71-74, 2014.

[8] Tomovic S., Radonjic M., Pejanovic Djurisic M., Radusinovic I., "Software Defined Wireless Sensor Networks", 20th Conference on Information Technologies IT 15, Zabljak, Montenegro, 2015.

[9] Z. Han, R. Wanli, "A novel wireless sensor networks structure based on the sdn." International Journal of Distributed Sensor Networks 2014 (2014).

[10] W. R. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” 33rd Hawaii Int’l. Conf. Sys. Sci., Jan. 2000.

[11] OpenFlow switch specification v1.0.0, Retrieved: Sept. 2015, Available at: http://archive.openflow.org/documents/openflow-spec-v1.0.0.pdf

[12] S. Costanzo, L. Galluccio, G. Morabito, and S. Palazzo, "Software defined wireless networks: Unbridling sdns," EWSDN, pp. 1–6, 2012.

[13] E. W. Dijkstra, "A Note on Two Problems in Connection with Graphs," in Numerische Math, vol. 1, pp. 269-271, 1959.

[14] W. B. Heinzelman, A. P. Chandrakasan and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," IEEE Transaction on wireless communications, vol.1, no.4, pp.660-670, Oct. 2002.

[15] OpenMote, www.openmote.com.

[16] S. Tomovic and I. Radusinovic, “Performance analysis of a new SDN-based WSN architecture,” *Telecommunications Forum Telfor (TELFOR), 2015 23rd*, Belgrade, 2015, pp. 99-102.