Data Collection From Isolated Clusters in Wireless Sensor Networks Using Mobile Ferries

Konstantin Mikhaylov and Jouni Tervonen
RFMedia Laboratory
Oulu Southern Institute,University of Oulu
Ylivieska, Finland
{konstantin.mikhaylov, jouni.tervonen}@oulu.fi

***Abstract*— The usage of mobile ferries, under which we understand the nodes that transfer the data between other nodes in process of their movement, provides multiple benefits for Wireless Sensor Networks (WSNs). In the paper, we suggest and evaluate several different approaches and protocols for collecting the data using such ferries from isolated WSN subnetworks for different scenarios. We present the results of the simulations that reveal the effects of the WSN density, ferry speed, node sleep policy and maximum hop limitation on the performance and required resources for the tested protocols.**

Keywords- wireless sensor networks; mobility; WSN; mobile ferry; MULE; data collection; routing; performance;

I. INTRODUCTION

The Wireless Sensor Networks (WSNs) have nowadays found numerous applications and are becoming an important part of everyday life. The contemporary WSNs have the tendency to get more and more heterogeneous and can unite within the same network the nodes with different architectures and capabilities [1], [2]. One of the potentially beneficial but very challenging scenarios is the presence of the mobile WSN nodes.

The mobility of the nodes in a WSN can be classified using three major terms: the mobility subject (i.e., what nodes are moving); the mobility nature (i.e., why the node is moving and who is controlling its movement) and the available data about mobility.

Depending on application, in the WSNs the subjects of mobility can be the sinks (i.e., the nodes that collect the data from the other nodes [3], [4]), the regular sensor nodes or the special service nodes that are intended to transfer the data between the other nodes (these nodes are usually called ferries or relays depending on their operation) [3], [5], [6].

As has been shown in the previous researches [7]–[9], the sink mobility can significantly increase network lifetime, reduce the traffic and decrease the communication latency. Nonetheless, mobile sinks usually require full control about their movement and cannot use wired interfaces for data exchange or power supply.

Under mobile relays, we understand the nodes that interconnect two or more isolated subnetworks in a WSN (i.e., the parts of the network that cannot communicate using e.g., packet relaying; in literature those are also referred to as WSN islands [10]) *by placing themselves between the isolated subnetworks and relaying all the data traffic between those subnetworks*. Naturally, the relays must fully control their movement and a single relay can connect only the subnetworks that have the nodes located within the communication range of the relay node R_{max} (see Fig. 1). As has been shown e.g., in [11], for some scenarios, the mobile relays significantly increase the lifetime of the WSN.

The mobile ferries (MFs) (those are also referred to as MULEs ([12], [13]), mobile data collectors ([14], [15]) or mobile elements ([3]) in literature) are the nodes that transfer the data between the isolated parts of the WSN *in the process of their movement with temporal data storage on-ferry* [13]. The most significant difference between the mobile sinks or mobile relays and the MFs is that the MF mobility does not

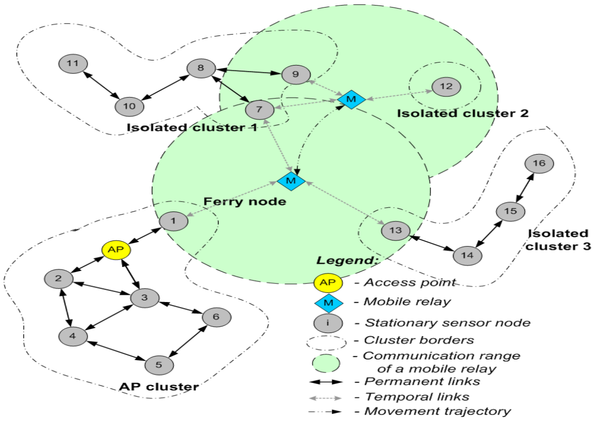

Figure 1. Example of the WSN with mobile relays.

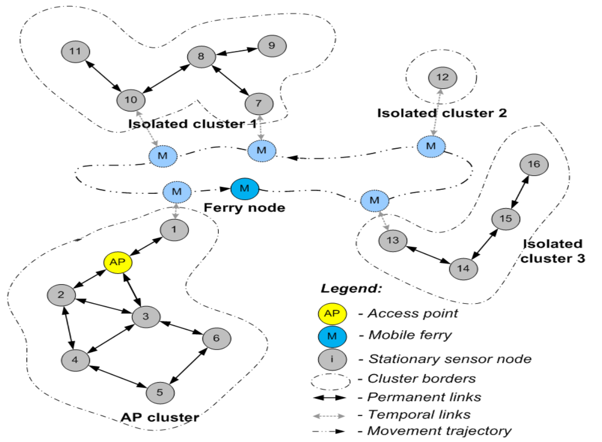

Figure 2. Example of the WSN with mobile ferries.

need to be controlled, while the relays and sinks usually require full control over their mobility. The other advantage of MFs is that those can be used to interconnect even distant subnetworks – see Fig. 2. As has been shown e.g. in [13], the MFs can increase the coverage of the WSN by connecting the sink and isolated subnetworks and improve the energy efficiency. The major disadvantage of the data transferring using MFs is the communication latency that depends on the network layout, speed and mobility pattern of the MFs.

In the paper, we focus on the MFs scenario and the data collecting from the isolated nodes and subnetworks using a MF. Below we suggest and evaluate several different data collection (DCol) protocols, under which we mean the combination of isolated cluster discovery, connection establishment, routing, and data communication mechanisms.

II. DATA COLLECTION FROM WSNs WITH MOBILE FERRY

The operation of the MFs typically includes three major phases:

- Isolated node or isolated clusters detection and connection establishment;
- Data download from the isolated nodes or clusters to the memory of the MF;
- Sink cluster detection, connection to it and data uploading from the MF memory to the sink.

In the case if the movement pattern of the MF is known in advance by all the nodes and the locations of all the nodes are known to the MF, the cluster detection and the connection establishment can be implemented effectively with the minimum energy consumption (consider e.g., [3], [16]). Nonetheless, if the MF movement schedule is unknown or if the route is subject to change, the nodes are unable to predict the time when the MF will arrive. In such scenario two following techniques can be used. The first option is to use the mechanism that wakes up the sensor node once the MF has arrived, e.g., the radio triggered activation [3] or activation from the sensor signal (e.g., button press or sensor data analysis). This solution ensures minimum energy consumption, but increases the price of the WSN node due to the external components use. The other option, which does not need any external components but causes higher energy consumption is the periodic listening by the WSN nodes for the advertisements issued by the MF during its movement [3], [12], [17].

Once the MF and the isolated nodes have detected each other, the MF can start leaching the data from the sensor nodes. In multiple researches considering the MF WSNs scenario the authors used the MF to collect the data only from nodes within the single-hop distance from the MF (consider, e.g. [17], [18]). Nonetheless, some of the authors have tried to utilize the existing routing protocols or suggested the new ones to establish the multi-hop route between the MF and the sensor nodes in the cluster. E.g., in [18] authors have used the Directed diffusion (DD) for collecting the data by MF. In [19] was presented the DCol strategy for the MFs moving along the predefined path. In [20] was introduced the simple network protocol that allows the MF to initiate the formation of the sub-network for each isolated cluster and to obtain the data from it. In [4] the authors exploit the information theoretic approach to developing the Weighted Entropy DAta diSsemination (WEDAS) protocol. In WEDAS, the forwarding nodes are selected basing on the remaining energy and the estimated position of the mobile element.

For discovering the sink cluster and transferring the data from the MF to the sink are usually used the same mechanisms as for the isolated clusters discovery, therefore we do not discuss those specifically.

III. NETWORK SCENARIO AND DATA COLLECTION PROTOCOLS FOR NETWORKS WITH MOBILE FERRIES

For investigating the different WSN DCol methodologies using the MF we have chosen the most general scenario. We assume that n static wireless sensor nodes are randomly placed in the area of (x, y) meters and their position is unknown. Each node has N data packets that should be forwarded to the sink node. At random moment of time, single MF approaches the test area and crosses it via some unpredictable route. The mobility of the MF cannot be controlled by the WSN in any way. For increasing the operation time of the WSN, the sensor nodes are allowed to use the low-power sleep mode. The sensors are not synchronized and their sleep schedules are independent.

Inputs:
$Addr_{host}$ – address of host node (i.e., next node on the route to ferry)
$Addr_{token\ holder}$ – address of the node that holds the token
$ID_{last\ beacon}$ –ID for the last received beacon packet
$ID_{last\ rqst}$ –ID for the last received token request packet
$ID_{last\ token}$ –ID for the last received token advertise packet
N_{buf} – number of packets in buffer on Sensors
N_{ferry} – distance from Ferry (number of hops)
nodeType – type of the node, can be either Ferry or Sensor
T_{beacon} – period for beacons (or token advertisement) transmitted by Ferry
T_{dataTX} – period of data transmission by Sensors
T_{rand} – random delay
Types of radio packets:
ACKP –acknowledgement packet
BP – beacon packet
DP – data packet
Tadv – token advertise packet
Trqst – token request packet
Tgrant – token grant packet
Treturn – token return packet
Radio packet arguments:
Data – actual data
ID – unique identification of this packet
ID_{ack} – unique identification of packet to be acknowledged
N_{hops} – number of hops the packet had already passed
N_{left} – number of hops left for this packet

Figure 3. Input data and radio packet format for the developed algorithms

```
if nodeType is Ferry
    broadcast $BP(ID_{ack})$ every $T_{beacon}$ seconds
    if $DP(ID, Data)$ received
        $ID \rightarrow ID_{ack}$
    end if
else if nodeType is Sensor
    if $BP(ID_{ack})$ received
        delete packet with $ID=IDack$ from buffer if it exists
        send next data packet from the buffer after $random(0, T_{rand})$ seconds
    end if
end if
```

Figure 4. Algorithm for data collection from single-hop (SH) neighbours

```
if nodeType is Ferry
    broadcast $BP(ID_{ack}, ID)$ every $T_{beacon}$ seconds
    if $DP(ID, Data)$ received
        $ID \rightarrow ID_{ack}$
    end if
else if nodeType is Sensor
    if $BP(ID_{ack}, ID)$ received and $ID \neq ID_{last\ beacon}$
        $ID \rightarrow ID_{last\ beacon}$
        address of $BP(ID_{ack})$ transmitter $\rightarrow Addr_{host}$
        if packet with $ID=ID_{ack}$ exists in buffer
            if packet with $ID$ was generated not by me
                send $ACKP(ID_{ack})$ to the node from which it has been received
            end if
            delete packet $ID_{ack}$ from buffer
            $N_{buf} - 1 \rightarrow N_{buf}$
        end if
        if $N_{buf} > 0$
            send next data packet from the buffer to $Addr_{host}$ after $random(0, T_{rand})$ seconds
        else
            rebroadcast $BP(ID_{ack}, ID)$ after $random(0, T_{rand})$ seconds
        end if
    else if $DP(ID, Data)$ received
        put $DP(ID, Data)$ packet in my data buffer
        $N_{buf} + 1 \rightarrow N_{buf}$
    else if $ACKP(ID_{ack})$ received
        if packet with $ID=ID_{ack}$ exists in buffer
            if packet with $ID$ was generated not by me
                send $ACKP(ID_{ack})$ to the node from which it has been received
            end if
            delete packet $ID_{ack}$ from buffer
            $N_{buf} - 1 \rightarrow N_{buf}$
        end if
    end if
end if
```

Figure 5. Algorithm for limited multi-hop single route (MSR) data collection

```
if nodeType is Ferry
    broadcast $BP(ID, N_{hops}=0)$ every $T_{beacon}$ seconds
    if $DP(ID, Data)$ received
        $ID \rightarrow ID_{ack}$
        send $ACKP(ID_{ack})$ to source of $DP(ID, Data)$
    end if
else if nodeType is Sensor
    if $BP(ID, N_{hops})$ received and $ID \neq ID_{last\ beacon}$
        $ID \rightarrow ID_{last\ beacon}$
        $N_{hops} \rightarrow N_{ferry}$
        rebroadcast $BP(ID, N_{Ferry}+1)$ after $random(0, T_{rand})$ seconds
        start broadcasting data packets $DP(ID, Data, N_{left})$ from buffer every $T_{dataTX}$ seconds
    else if $DP(ID, Data, N_{left})$ received and $N_{ferry} < N_{left}$
        $ID \rightarrow ID_{ack}$
        put $DP(ID, Data)$ packet in my data buffer
        send $ACKP(ID_{ack})$ to source of $DP(ID, Data, N_{left})$
    else if $ACKP(ID_{ack})$ received
        delete packet $ID_{ack}$ from buffer
    end if
end if
```

Figure 6. Algorithm for data collection using unlimited multi-hop multiroute flooding (MMF) towards ferry

For such scenario, the main task for the data collecting protocol is the reliable (i.e., without packet losses) data transfer from the sensor nodes to the MF. Unfortunately, the majority of the existing protocols cannot be used for this scenario due to a high level of input data uncertainty. Therefore, below we suggest the five protocols for collecting the data from such environment. The algorithms describing the protocols are presented in Figs. 4-8 with Fig. 3 revealing the designation used by all algorithms.

The first algorithm (see Fig. 4) implements the most basic data collecting protocol for the MF scenario and is similar to the one used e.g., in [17], [18]. This protocol implies the periodic broadcasting of the beacon packet by the MF to advertise its presence to the sensor nodes. The WSN nodes within the communication range of the MF that receive this advertisement, reply with the data packet. To improve the reliability of the communication, we have included in the beacon packet the ID_{ack} field that signalizes the sensor node that its previous data packet has been successfully received by MF.

```
if nodeType is Ferry
    if $Addr_{token\ holder}$ = myself
        broadcast $Tadv(ID)$ every $T_{beacon}$ seconds
    endif
    if $Trqst(ID)$ is received and $Addr_{token\ holder}$ = myself
        initial source of $Trqst(ID) \rightarrow Addr_{token\ holder}$
        send $Tgrant()$ to the source of $Trqst(ID)$ and start "connection lost" timer
    else if $DP(ID, Data)$ received
        rebroadcast the message (used instead of ACK) and restart "connection lost" timer
    end if
    if "connection lost" timer fired or $Treturn()$ is received and stop "connection lost" timer
        myself $\rightarrow Addr_{token\ holder}$
    endif
else if nodeType is Sensor
    if $Tadv(ID)$ is received and $ID \neq ID_{last\ token}$
        ferry $\rightarrow Addr_{token\ holder}$
        if $N_{buf} > 0$
            address of $Tadv(ID)$ transmitter $\rightarrow Addr_{host}$
            send $Trqst(ID)$ to $Addr_{host}$ after $random(0, T_{rand})$ seconds
        else
            rebroadcast $Tadv(ID)$ after $random(0, T_{rand})$ seconds
        endif
    else if $Trqst(ID)$ is received and $ID \neq ID_{last\ rqst}$
        $ID \rightarrow ID_{last\ rqst}$
        forward $Trqst(ID)$ to $Addr_{host}$
    else if $Tgrant(Addr_{token\ holder})$ is received
        address of $Tgrant(Addr_{token\ holder})$ transmitter $\rightarrow Addr_{host}$
        if $Addr_{token\ holder}$ = myself
            start sending $DP(ID, Data)$ to $Addr_{host}$ every $T_{dataTX}$ seconds
        else
            rebroadcast $Tgrant(Addr_{token\ holder})$ after $random(0, T_{rand})$ seconds
        endif
    else if $Treturn()$ is received
        forward $Treturn()$ to $Addr_{host}$
    else if $DP(ID, Data)$ is received
        if $DP(ID, Data)$ is for me
            retransmit $DP(ID, Data)$ to $Addr_{host}$
        else if $DP(ID, Data)$ is from $Addr_{host}$
            delete packet $ID$ from buffer
            $N_{buf} - 1 \rightarrow N_{buf}$
            if $N_{buf}=0$
                send $Treturn()$ to $Addr_{host}$
            endif
        endif
    end if
end if
```

Figure 7. Algorithm for limited token-based data leaching (TDL)

```
if nodeType is Ferry
    broadcast $BP(ID_{beacon,})$ every $T_{beacon}$ seconds
    if $DP(ID, Data)$ received
        $ID \rightarrow ID_{ack}$
        send $ACKP(ID_{ack})$ to source of $DP(ID, Data)$
    end if
else if nodeType is Sensor
    if $BP(ID)$ received and $ID \neq ID_{last\ beacon}$
        address of $BP(ID)$ transmitter $\rightarrow Addr_{host}$
        $ID \rightarrow ID_{last\ beacon}$
        move one data packet from buffer to bufferII
        rebroadcast $BP(ID_{beacon})$ after $random(0, T_{rand})$ seconds
        send data packet from bufferII to $Addr_{host}$ after $T_{dataTX}$ seconds and start "packet lost" time
    else if $ACKP(ID_{ack})$ received
        delete packet $ID_{ack}$ from bufferII
        if bufferII is not empty
            send data packet from bufferII to $Addr_{host}$ after $T_{dataTX}$ seconds and start "packet lost" time
        endif
    else if $DP(ID, Data)$ received
        $ID \rightarrow ID_{ack}$
        put $DP(ID, Data)$ packet in my data bufferII
        send $ACKP(ID_{ack})$ to source of $DP(ID, Data)$
        send data packet from bufferII to $Addr_{host}$ after $T_{dataTX}$ seconds and start "packet lost" timer
    end if
    if "packet lost" time had fired
        send data packet from buffer to $Addr_{host}$ after $T_{dataTX}$ seconds and restart "packet lost" timer
    end if
end if
```

Figure 8. Algorithm for collecting a single data packet from all the nodes in the network (SPC)

Unlike the first algorithm, the second and the following enable the use of multi-hop forwarding. E.g., for the algorithm presented on Fig. 5 the WSN nodes once finished transmitting all data packets, start to rebroadcast the beacon advertisements and forward the data traffic from the other nodes to the MF. The data from the sensor node to the MF are transmitted via the route with the minimum transmission time for the beacon packet from the MF to this sensor. In case of data or acknowledgement packets collision – the data is rebroadcasted once again after the next beacon.

The algorithm presented in Fig. 6 uses the multi-hop multi-route flooding technique that has been suggested for partially-connected WSNs. Once receiving the beacon, the sensor nodes rebroadcast it and start sending data packets. The neighbor nodes that are located closer to the MF (i.e., having less hops) acknowledge, save and forward the packets from the more distant nodes.

The algorithm presented in Fig. 7 utilizes the token mechanism. The MF advertises the token, for which the sensor nodes compete. The winner – transmits all its packets to the MF with short delay. In case if the node has no data to transmit, it retransmits the token advertisement further enabling the other nodes to enter the competition. Currently, only one node can hold the token at a time. The "connection lost" timer on the MF ensures network recovery in the case if the token has been lost or if data connections have been broken.

Finally, the Single-Packet Collect (SPC) algorithm presented in Fig. 8 tries to collect only a single packet but from each node in the WSN. This protocol is especially useful for the applications that require the "snapshot" of the measurements in the whole WSN and do not care much about the measurements' history for specific nodes. To the best of our knowledge, this is the first protocol for WSNs that tries to equalize the DCol from different sensor nodes. For distinguishing the traffic from the other nodes that should be transmitted as soon as possible from node's own packets, the second buffer (bufferII) has been introduced.

IV. EVALUATION METHODOLOGY AND RESULTS

For evaluating the described protocols, all those have been implemented in MiXiM [21] framework of OMNeT++ [22] network simulator. During the simulation, we randomly placed *n* nodes over the area of 500 m by 500 m. For the radio we have used the model of CC2430 radio chip from Texas Instruments. The MF, equipped with the same radio module, was crossing the test area diagonally. All the radios on the nodes used 250kbit/s datarate and the constant transmit power of 0dBm (giving maximum communication range around 120 meters). The radio channel access has been implemented using the clear channel assessment (CCA) technique as described in IEEE 802.15.4.

At the beginning of simulation, 25 radio packets of 14 data-bytes each were placed in the memory on each node. For the simulations, we assume that each node has unlimited memory to store its own and forwarded packets and no packets are dropped due to buffer overflow. Nonetheless, as this assumption is far from the truth for the real-life, we have also monitored the number of data packets stored on WSN nodes over time to estimate the required resources and the scalability of the protocols.

In the simulations, we have investigated the effect of the following parameters on the performance of the algorithms (estimated basing on the number of successfully delivered unique data packets to the MF): the density of the nodes; the policy for the sensor node sleep mode; the MF speed and the maximum hop limitation. The results of the simulations revealing these effects are presented in Figs. 9-18. The data presented in the charts represent the averaged value over at least 20 runs for different network layouts.

The tests for evaluating the effect of the WSN node density on the protocol operation were executed for 10 to 150 nodes randomly placed in the test area. During the test, the MF was moving with the constant speed of 10 m/s and all the sensor nodes were in active mode. The required for the protocols parameters were specified as follows: $T_{beacon(SH)}=T_{beacon(MSR)}=0.05s$, $T_{beacon(MMF)}=0.1s$, $T_{beacon(TDL)}=T_{beacon(SPC)}=0.5s$; $T_{dataTX(MMF)}=0.0125s$, $T_{dataTX(TDL)}=T_{dataTX(SPC)}=0.01s$; $T_{rand}=0.01s$. Besides, for token-based protocol the "connection lost" timer was set equal to $T_{beacon(TDL)}$.

As revealed in Fig. 9 the SH protocol managed to collect around 57% of the generated packets (which is close to theoretically expected – as the MF on its way had 56.3% of the test area within its communication range). The multi-hop protocols allowed to increase the amount of collected packets to almost 100% (for TDL and SPC) for dense-enough network. The MSR and MMF protocols worked quite well for the sparse network scenario, but for the dense network their efficiency fell below TDL and SPC due to high network traffic (see Fig. 10). This is especially notable for MMF protocol – due to constant collisions, its performance fall well below SH. For measuring the regularity of packet reception from different nodes we have used Pearson's cumulative test statistic (see Equation 7.1 in [23]) assuming the uniform distribution of received packets' sources as null hypothesis. As can be seen from Fig. 11, for all the protocols except SPC, the MF gets the data packets from different sources very irregularly (the higher the value is – the more irregularly the MF received the packets). For SPC, the packets from different nodes are delivered much more regularly. As revealed in Fig. 12, for SH, MSR and TDL protocols the WSN nodes didn't need extra memory for storing forwarded packets, while for SPC and especially for MMF the memory buffer size increased with the increase of the total node number in the network.

For investigating the effect of the MF speed, we have repeated the described above tests for various speeds of the MF for 20 different layouts of 50-node WSN. During the tests, all the WSN nodes were in active mode. The values of $T_{beacon(TDL)}$ and $T_{beacon(SPC)}$ have been reduced to 0.1s to equalize the test conditions. As revealed in Fig. 13 for low MF speed, the multi-hop DCol protocols (TDL, SPC, MMF) outperform the single-hop protocol, but for high speed – the difference in the performances of protocols is minimal. Furthermore, the performance of SPC protocol for high MF speed is even below that of SH protocol.

The dependences presented in Fig. 14 reveal the effect of the sleep mode on the performance of the DCol protocols. During the sleep mode tests, we have used the same 50-node WSNs and the same parameters of protocols as during speed-

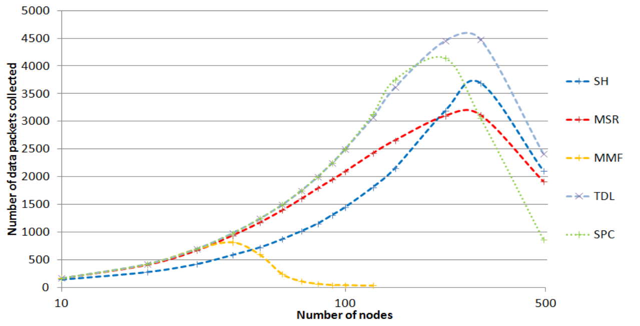

Figure 9. Effect of nodes density on the amount of collected packets for tested protocols (total generated packets = 25 x number of sensor nodes).

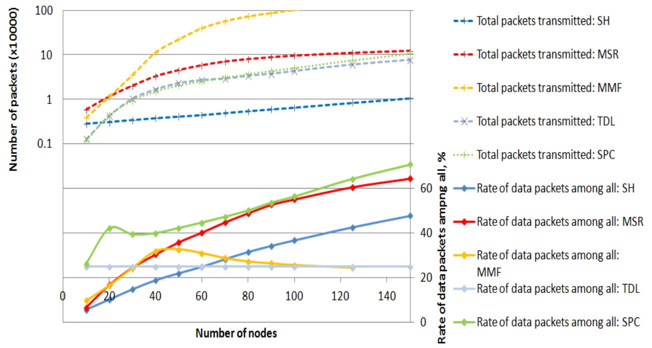

Figure 10. Effect of nodes density on network traffic (i.e., total number of transmitted packets)

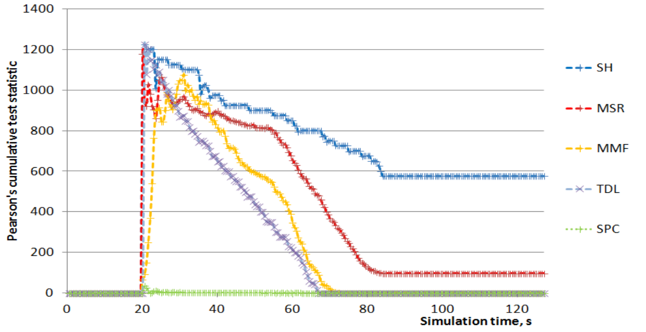

Figure 11. Pearson's cumulative test statistic (uniformal distribution assumed) for distribution of the received packet's sources over time for different protocols for example WSN layout (50 nodes)

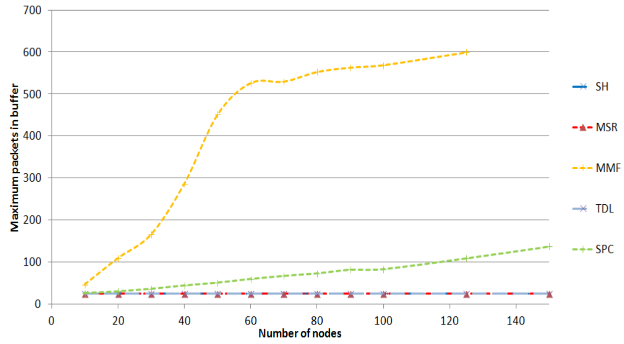

Figure 12. Effect of nodes density on maximum number of packets in sensor node buffers

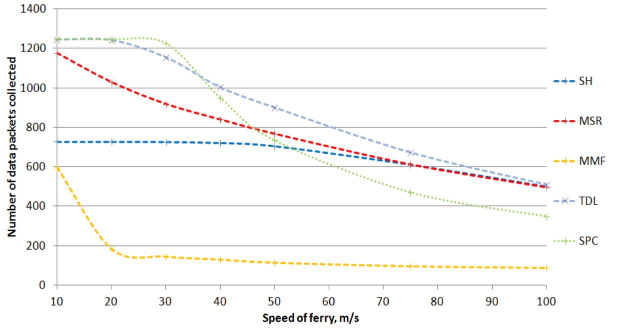

Figure 13. Effect of ferry's speed on the performance of data collection protocols (totall number of generated packets: 1250)

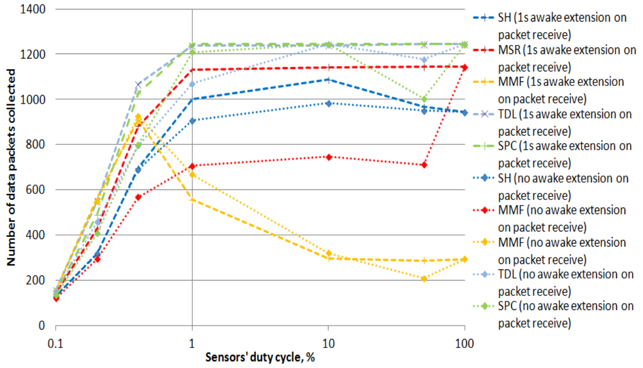

Figure 14. Effect of sensors' sleep mode on data collection algorithms' performance (totall number of generated packets: 1250)

tests, but the MF had a constant speed of 10 m/s. During the initialization, each sensor node independently specified its sleep schedule using the specified duty cycle. The length of the stay-awake period in all the tests was set to 0.1s, while the wake-up period varied from 0.1s (node is awake all the time) and up to 100s (node is awake 0.1% of time). We assumed that at the beginning of simulation each sensor had 2 mAh (23.76 J) of energy and monitored the change of nodes' energy for different protocols and settings (using the energy framework presented in [24]). During this test we have also evaluated the effect of using the traffic from other nodes to alert the sensors about the MF approach – i.e., in the first scenario sensor nodes extended their awake period only if they receive the beacon packet, while in second scenario each correctly received data packet (even not intended for this node) resulted in the increase of the awake-period by one second. As revealed in Fig. 14 the extension of the awake period at the reception of the data from the other nodes allowed to significantly increase the number of the collected packet for low node duty cycles - using TDL and SPC and sensor duty cycles above 1% the MF had collected almost all packets without significant reduction in total energy consumption (see Figs. 14 and 15). Both the performance and energy consumption reduction for MMF protocol without wake-up period extension were caused by the packet collisions: due to collisions the remote nodes were not getting the beacon packets from the MF and were switching back to the sleep mode.

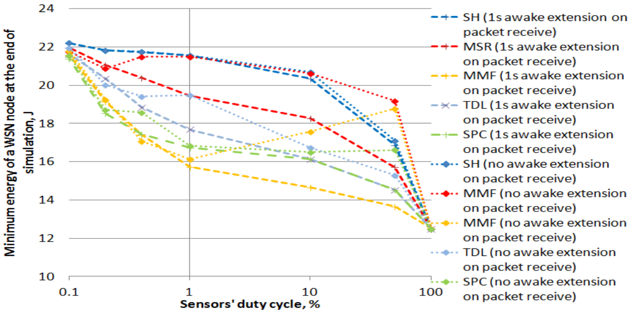

Figure 15. Effect of sensors' sleep mode on energy consumption (energy of nodes at start of simulation: 23.76 J)

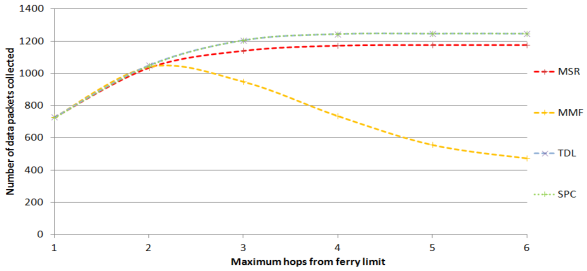

Figure 16. Effect of hop limitation on data collection algorithms' performance (totally 1250 packets were generated)

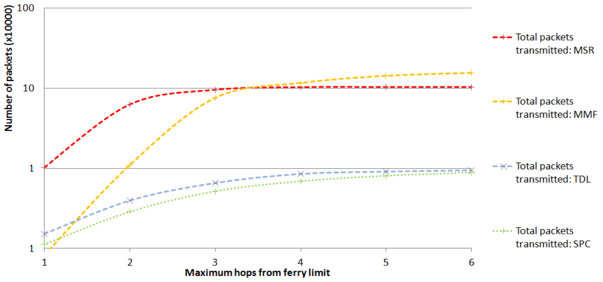

Figure 17. Effect of hop limitation on network data traffic.

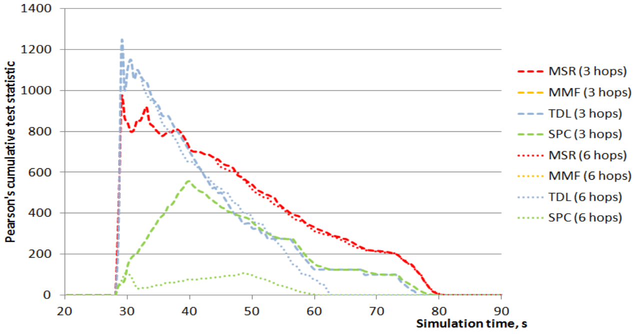

Figure 18. Effect of hop limitation on distribution of the received packet's sources.

Finally, Fig. 16 reveals the effect of limiting the maximum number of hops for the tested DCol protocols. In all previous tests, the protocols were not limited in the length of the used route. For the current test, we intentionally limited the maximum route length between the MF and the sensor nodes. For the tests, we used 50-node WSN with all the nodes in active mode. As revealed in Figs. 16 and 17, for the tested scenario the hop number limitation allowed to reduce the network data traffic without the loss of performance at a cost of increasing the time required to collect the packets (see Fig. 18). For MMF protocol the hop number limitation to 2 allowed to solve the over-flooding problem and significantly increase the performance of the algorithm. Nonetheless, this result has been obtained for the scenario when the MF was crossing the sensor distribution area via the best route. For the other scenarios, the hop number limitation could cause the fail of packet delivery from the distant nodes.

V. DISCUSSION AND CONCLUSIONS

In the current paper, we have presented and evaluated five different data-collection strategies that allow the mobile ferry to collect the data from the isolated WSN subnetworks. The tested data-collection protocols are for the most general scenario – when no information about ferry movement pattern or any location data are available. The suggested in the paper Single-Packet Collect (SPC) protocol is the first protocol, to the best of our knowledge, which targets the equability of data packet acquisition from different WSN nodes. We consider that data collection protocols of such sort are especially beneficial for the ferry-based data collection scenario, as such protocols provide the single or several measurement values from the maximum number of sensor nodes in minimum time.

As has been shown in the paper, for the relatively sparse WSN, the multi-hop data collection protocols can get more packets than single-hop ones, but with the increase in the node density or with the increase in the ferry movement speed the performance for multi-hop data collection protocols reduces due to the packet collisions and some of multi-hop protocols become less effective than the single-hop data collection protocol. The presented simulation results revealed that even in the WSN with asynchronous node sleep schedules and low duty cycles, the ferry can effectively collect the data. Due to sporadic data transfer for the MF scenario (i.e., data are transmitted only when the ferry approaches the isolated cluster), the extension of the awake period for the side-traffic reception (i.e., the reception of service and data packets that are not intended for this node) allowed to increase the number of the collected packets at a cost of a minor increase in the consumed energy. For the protocol based on multi-route data flooding, the limitation of the maximum number of hops between the ferry and the sensors allowed to partially solve the over-flood problem, nonetheless its efficiency was still quite low.

Based on the presented in the paper data we can draw the following conclusions. The common single-hop data collecting protocol is as effective as the multi-hop protocols in three cases: a very dense network; a very high speed of ferry or a very low node duty cycle. In other scenarios, the multi-hop protocols will usually have higher performance. As has been shown in the paper, the use of multi-route flooding for the selected scenario gave very poor results due to extremely high network traffic and packet collisions. The use of the single route limited flooding (MSR) provided somewhat better results although for high network density its performance was rather poor. Among the tested protocosl, the maximum number of packets was collected by the suggested in the paper token-based data leaching (TDL) and single-packet collect (SPC) protocols. The former one switches between the nodes, leaching all the data from each node's buffer. The latter one tries to collect the data from different nodes uniformly thus allowing to get a packet from the maximum number of different nodes in minimum time.

In the future, we are planning to extend the presented simulations by also evaluating the different ferry movement scenarios. Besides, we are planning to implement the TDL and SPC protocols in hardware and test those in real-life using the real-life mobile ferry hardware platform.

REFERENCES

[1] K. Mikhaylov and J. Tervonen, "Node's power source type identification in wireless sensor networks," Proc. International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA'11), Oct. 2011, pp. 521–525, doi:10.1109/BWCCA.2011.84.

[2] K. Mikhaylov and J. Tervonen, "Energy–efficient routing in wireless sensor networks using power–source type identification," Int. J. of Space-Based and Situated Computing, vol. 2, no. 4, Oct. 2012, pp. 253–266, doi:10.1504/IJSSC.2012.050008.

[3] M. Di Francesco, S. K. Das, and G. Anastasi, "Data collection in wireless sensor networks with mobile elements: a survey," ACM Trans. Sen. Netw., vol. 8, no. 1, Aug. 2011, pp. 7:1–7:31, doi:10.1145/1993042.1993049.

[4] H. M. Ammari and S. K. Das, "Data dissemination to mobile sinks in wireless sensor networks: an information theoretic approach," Proc. IEEE International Conference on Mobile Adhoc and Sensor Systems Conference (MAHSS'05), IEEE Press, Nov. 2005, pp. 306–314, doi:10.1109/MAHSS.2005.1542814.

[5] L. He, Y. Zhuang, J. Pan, and J. Xu, "Evaluating on-demand data collection with mobile elements in wireless sensor networks," Proc. IEEE Vehicular Technology Conference (VTC'10-Fal), IEEE Press, Sep. 2010, pp. 1–5, doi:10.1109/VETECF.2010.5594515.

[6] W. Jiao, L. Cheng, M. Chen, C. Chen, and J. Ma, "Efficient data delivery in wireless sensor networks with ubiquitous mobile data collectors," Proc. IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC'10), IEEE Press, Dec. 2010, pp. 232–239, doi:10.1109/EUC.2010.41.

[7] M. Marta and M. Cardei, "Using sink mobility to increase wireless sensor networks lifetime," Proc. International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM'08), IEEE Press, Jun. 2008, pp. 1–10, doi:10.1109/WOWMOM.2008.4594857.

[8] S. Sharma, M. Rani, and S. B. Goyal, "Energy efficient data dissemination with ATIM window and dynamic sink in wireless sensor networks," Proc. International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom'09), IEEE Press, Oct. 2009, pp. 559–564, doi:10.1109/ARTCom.2009.9.

[9] S. Tang et al., "DAWN: energy efficient data aggregation in WSN with mobile sinks," Proc. 18th International Workshop on Quality of Service (IWQoS'10), IEEE Press, Jun. 2010, pp. 1–9, doi:10.1109/IWQoS.2010.5542765.

[10] G. Pantziou, A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and B. Mamalis, "Mobile ainks for information retrieval from cluster-based WSN islands," Proc. International Conference on Ad Hoc Networks and Wireless (ADHOC-NOW'09), Springer-Verlag, 2009, pp. 213–226, doi:10.1007/978-3-642-04383-3_16.

[11] W. Wang, V. Srinivasan, and K. Chua, "Using mobile relays to prolong the lifetime of wireless sensor networks," Proc. International Conference on Mobile Computing and Networking (MobiCom'05), ACM, 2005, pp. 270–283, doi:10.1145/1080829.1080858.

[12] S. Basagni, A. Carosi, and C. Petrioli, "Controlled vs. uncontrolled mobility in wireless sensor networks: some performance insights," Proc. IEEE Vehicular Technology Conference (VETECF'07 Fall), Mar. 2007, pp. 269–273, doi:10.1109/VETECF.2007.70.

[13] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: modeling a three-tier architecture for sparse sensor networks," Proc. IEEE Workshop on Sensor Network Protocols and Applications (SNPA'03), IEEE Press, May 2003, pp. 30–41, doi:10.1109/SNPA.2003.1203354.

[14] T. Chen, T. Chen, and P. Wu, "Data collection in wireless sensor networks assisted by mobile collector," Proc. IFIP Wireless Days (WD'08), IEEE Press, Nov. 2008, pp. 1–5, doi:10.1109/WD.2008.4812895.

[15] M. Zhao and Y. Yang, "Data gathering in wireless sensor networks with multiple mobile collectors and SDMA technique sensor networks," Proc. IEEE Wireless Communications and Networking Conference (WCNC'10), IEEE Press, Apr. 2010, pp. 1–6, doi:10.1109/WCNC.2010.5506199.

[16] H. Jun, W. Zhao, M. H. Ammar, E. W. Zegura, and C. Lee, "Trading latency for energy in densely deployed wireless ad hoc networks using message ferrying," Ad Hoc Netw., vol. 5, no. 4, May 2007, pp. 444–461, doi:10.1016/j.adhoc.2006.02.001.

[17] G. Anastasi, M. Conti, E. Gregori, C. Spagoni, and G. Valente, "Motes sensor networks in dynamic scenarios: an experimental study for pervasive applications in urban environments," International Journal of Ubiquitous Computing and Intelligence, vol. 1, no. 1, Apr. 2007, pp. 9–16, doi:10.1166/juci.2007.002.

[18] A. Kansal, A. A. Somasundara, D. D. Jea, M. B. Srivastava, and D. Estrin, "Intelligent fluid infrastructure for embedded networks," Proc. International Conference on Mobile Systems, Applications, and Services (MobiSys '04), ACM, 2004, pp. 111–124, doi:10.1145/990064.990080.

[19] S. Gao, H. Zhang, and S. K. Das, "Efficient data collection in wireless sensor networks with path-constrained mobile sinks," IEEE Transactions on Mobile Computing, vol. 10, no. 4, Apr. 2011, pp. 592–608, doi:10.1109/TMC.2010.193.

[20] A. A. Somasundara, A. Kansal, D. D. Jea, D. Estrin, and M. B. Srivastava, "Controllably mobile infrastructure for low energy embedded networks," IEEE Trans. Mobile Computing, vol. 5, no. 8, Aug. 2006, pp. 958–973, doi:10.1109/TMC.2006.109.

[21] A. Kopke et al., "Simulating wireless and mobile networks in OMNeT++ the MiXiM vision," Proc. International Conference on Simulation Tools and Techniques (Simutools'08), ICST, 2008, pp. 71:1–71:8, doi:10.1145/1416222.1416302.

[22] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," Proc. International Conference on Simulation Tools and Techniques (Simutools '08), ICST, 2008, pp. 60:1–60:10, doi:10.1145/1416222.1416290.

[23] J. Tervonen, "Empirical modelling of the effects of tropospheric scintillation on radiowave propagation for satellite communications," PhD. Thesis, Helsinki University of Technology, Espoo, 1997.

[24] L. M. Feeney and D. Willkomm, "Energy Framework: an extensible framework for simulating battery consumption in wireless networks," Proc. International Conference on Simulation Tools and Techniques (SIMUTools'10), ICST, 2010, pp. 20:1–20:4, doi:10.4108/ICST.SIMUTOOLS2010.8725.