DS-MMAC: A Delay-Sensitive Multi-Channel MAC Protocol for Ambient Assistant Living Systems

Haiming Chen, Li Cui

Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

Abstract: In Ambient Assistant Living (AAL) systems, it is a fundamental problem to ensure prompt delivery of detected events, such as irregular heart rate or fall of elderly, to a central processing device (e.g. gateway node). Most of recently proposed MAC protocols for low-power embedded sensing systems (e.g. wireless sensor networks) are designed with energy efficiency as the first goal, so they are not suitable for AAL systems. Although some multi-channel MAC protocols have been proposed to address the problem, most of those protocols ignore the cost of channel switching, which can have reverse effect on network performance, especially latency of data delivery. In this paper, we propose a Delay-Sensitive Multi-channel MAC protocol (DS-MMAC) for AAL systems, which can provide high packet delivery ratio and bound low latency for data delivered to the gateway node. The novelty of the protocol is that an efficient distributed time slot scheduling and channel assignment algorithm is combined with the process of route establishment, which takes the channel switching cost into account and reduces end-to-end delay to meet the required delay bound of each data flow. The performance of the proposed protocol is evaluated through extensive simulations. Results show that DS-MMAC can bound low latency for delivering detected events in AAL system to the gateway, while providing high delivery reliability and low energy consumption.

Keywords: ambient assistant living; multi-channel mac; channel switching; slot scheduling; channel assignment

I. INTRODUCTION

Ambient intelligence technologies are developed with wide spread of embedded or wearable digital devices, such as comfort sensors to measure air condition, welfare and health devices to monitor physiological indexes, and unobtrusive sensing devices using sound or movement sensors to detect human activity[1]. They are applied in our daily life to make us have intelligent and natural interactions with the physical environments, through the use of information technology for real time monitoring and evaluation of critical data, triggering alarms and making recommendations. Ambient Assistant Living (AAL) is a new approach which exploits ambient intelligence to develop context-aware healthcare for older elderly or to construct “smart-homes” for aging-in-place, helping them maintain independent living[2]. To achieve this goal, AAL systems usually comprise a field of heterogeneous systems ranging from individual sensing devices to complex networked subsystems, which are deployed to cooperatively sense context and make reasonable decisions. Figure 1 illustrates a practical scenario of AAL applications.

Because most of detected events in AAL system are emergent, such as irregular heart rate or fall of elderly, which are required to be transmitted to a gateway node promptly, it is one of the fundamental problems to ensure low-latency communication among spatially distributed devices in AAL systems. Considering that the embedded or wearable digital devices used in AAL systems are usually equipped with low-power low-range wireless modules[3], they have similar network behavior with wireless sensor networks (WSN). Hereby, we give a brief overview on current solutions to address the problem of ensuring low-latency data transmission in wireless sensor networks.

1.1 Related work

Since sensor network usually runs in low duty cycle for energy efficiency, most of those protocols are designed to address issues raised from asynchronous sleep and wakeup among nodes. Those protocols can be divided into two categories. One employs scheduled listening, like SMAC[4] and TMAC[5], and the other uses low power listening, like BMAC[6] and XMAC[7]. They can perform well in networks with periodical or predictable data traffic, but cannot guarantee delivery latency for networks with unpredictable event-driven data traffic. Hence, a few of improved protocols were currently proposed for applications demanding prompt and reliable data transmission, such as RI-MAC[8] and DW-MAC[9].

Considering that the low power radio used in lightweight platforms of wireless sensor networks and AAL systems usually has low bandwidth, more and more researchers exploit the underlying support of multi-channel communication to fundamentally solve the problem. Recently, some multi-channel MAC protocols are proposed for WSN, e.g. MMSN[10], TMMAC[11], YMAC[12], TMCP[13], and PIP[14]. The aim of those protocols is to allow parallel data transmission, and to improve network throughput. However, most of those protocols ignore the cost of channel switching, which can have reverse effect on network performance, especially latency of data delivery.

In our previous work[15], we took the commonly used low-power RF module TI CC1000 [16] as the test object, and conducted extensive empirical studies to get three important parameters when designing multi-channel protocols for wireless sensor network. Those three important parameters are (1) number of available orthogonal channels, (2) time required to switch between two channels, and (3) total energy consumed to switch between two channels. Experimental results show that there are 48 channels available for the CC1000 based wireless nodes. The results related to the latter two parameters are shown in Table I.

As shown in Table I, channel switching is divided into three phases, which are calibrating receiving frequency (Cal-RX), calibrating transmitting frequency (Cal-TX), and restarting. From the above table, we can see that it costs the node about 50 milli-seconds and 1940 micro-joules to switch between channels. Referring to those results, we can conclude that when designing multi-channel communication scheme for low-power embedded computing systems, the cost of channel switching should be taken into consideration.

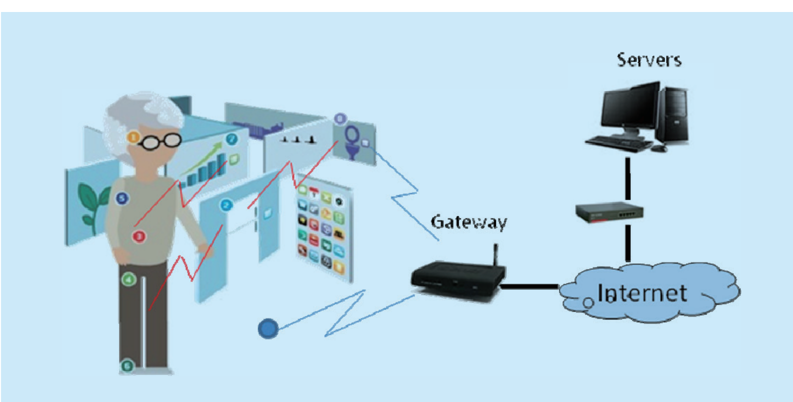

Fig.1 *A scenario of AAL applications*

Table I *Costs of channel switching*

Phrase	Current (mA)	Power (mW)	Duration (ms)	Energy (μJ)
Cal-RX	15.2	45.5	22.1	1005.55
Cal-TX	11.9	35.8	23.4	837.72
Re-start	7.5	22.4	4.3	96.32
Total	34.6	103.7	49.8	1939.59

1.2 Our contribution

In this paper, we propose a delay-sensitive multi-channel MAC protocol for AAL systems, named DS-MMAC, which ensures minimized delay for data transmission in the network. The main contributions are as follows.

(1) DS-MMAC is designed in a cross-layer approach. Channels are assigned along with route establishment, and an efficient distributed algorithm is proposed for reserving link that satisfies the required bandwidth and reduces the delay by a local scheduling scheme that minimizes one-hop delay.

(2) DS-MMAC inherits the property of energy efficiency from the Time Division Medium Access (TDMA). Nodes are only scheduled to wake up in the assigned slots, while keep sleeping in the idle slots to save power.

(3) The performance of the proposed protocol is evaluated through extensive simulations. Results show that DS-MMAC outperforms the widely used protocols in terms of packet delivery ratio, end-to-end delay and energy usage for each packet.

The rest of the paper is organized as follows. In section II, we formally describe the problem. In section III, we give a brief introduction on the protocol design and present the results of performance evaluation through simulation in section IV. At last, a conclusion is made in section V.

II. Problem Statement

Figure 1 shows a typical scenario of AAL applications, in which occurrence of a concerned event will trigger the embedded devices to collect data and send it to the gateway node. In such a scenario, the collected data are required to be received by the gateway node promptly and to be forwarded to the remote server reliably. It is difficult for the traditional low-power embedded network (e.g. wireless sensor network), which is composed of nodes with one radio module and sharing a wireless channel, to achieve this goal. With development of communication technology, each embedded sensing device is equipped with more than one radio modules [17], and can use multi-channels to communicate with each other. So we consider it is potential to exploit the multiple radios and multiple channels to accelerate delivery of detected events in AAL systems. However, it is a challenging problem to assign time slot and channel to a pair of radios at each hop, with the costs of channel switching shown in Table I taken into consideration, to ensure minimized delay for data transmission in the AAL system. In this section, we will formally describe the problem.

2.1 Network model

We use an undirected graph $G = (V, E)$ to model the network of an AAL system, where $V=\{v_1, v_2, \ldots, v_n\}$ is the set of embedded sensing devices and E is the set of undirected communication links. Hence, we think that each node v_i is equipped with $\mathcal{K}_i$ radios ($\mathcal{K}_i \geq 1$), and $\mathcal{R}_{vi}$ is the set of radios on node v_i, i.e., $\mathcal{R}_{vi}=\{R_{i1}, R_{i2}, \ldots, R_{ik_i}\}$. Each node may have different number of radios. In the system, totally w non-overlapping frequency channels are available, denoted by the set $\mathcal{F}=\{f_1, f_2, \ldots, f_w\}$. Because of local interference, each node should ensure that no more than one radio is active in the same channel simultaneously. Therefore, the number of radios v_i that can be used for communication at a node v_i is at most the number of channels w. The network model is illustrated by Figure 2, where the colored solid lines means the proper channels that have been assigned to the nodes at each hop, and

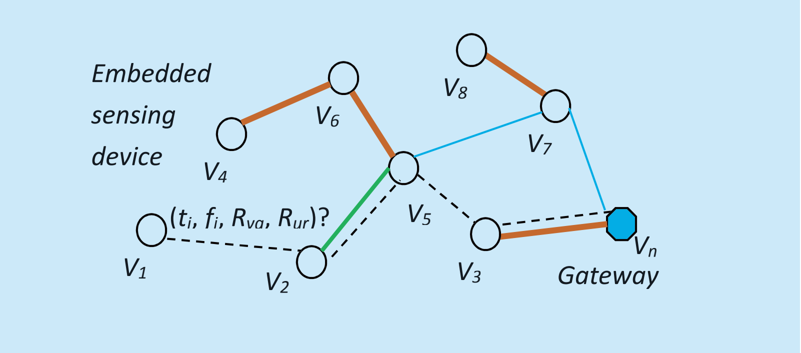

Fig.2 *An illustration of delay-sensitive multi-channel networking problem in AAL systems*

the dashed lines means the channels that are to be assigned by our proposed protocol. The notation ‘(t_i, f_j, R_{vq}, R_{ur}) ?’ beside the dashed line means which time slot and channel should be assigned to which radios of the nodes v_v and v_u to get minimized transmission delay of detected events in AAL system. It is essentially the problem to be addressed by this paper.

We assume that each node has the same transmission range R_T and uniform interference range R_I. So under the same channel the necessary condition for a node v_i to directly communicate with v_j successfully is $\|v_i-v_j\|\leq R_T$, where $\|v_i-v_j\|$ is the Euclidean distance between v_i and v_j. And a receiving node v_j is always interfered by the signal in the same channel from v_k if $\|v_j-v_k\|\leq R_I$, while v_k is not the sender of the communication to v_j. Typically it is assumed that $R_I \cong 2R_T$.

III. Protocol Design of DS-MMAC

In this section, we give a brief introduction of our cross-layer protocol design of DS-MMAC. The novelty of the protocol is that an efficient distributed algorithm is combined with the process of route establishment for link reservation, which takes the channel switching cost into account and reduces end-to-end delay to meet the required delay bound of each data flow.

3.1 Routing procedure

When an object or an emergent issue occurs in the network, the nearby nodes will serve as the source nodes. The source nodes will initiate the ROUTE REQUEST (RREQ) for a session. RREQ includes the following information: source-destination ID (s-d), the session ID, remaining latency bound (RLb) which is $\mathcal{D}$ originally, the requested capacity $\mathcal{B}$ for this session, TTL of the packet RREQ, reservation list (t_i, f_j, R_{vq}, R_{ur}), route list. TTL is used to do restrict flooding when broadcasting RREQ packets. Here TTL is $\delta+H_G(s, d)$, where $H_G(u, v)$ is the hop-distance between two nodes u and v in the communication graph G. The reservation list (t_i, f_j, R_{vq}, R_{ur}) means that the interface q of the node v_v and the interface r of the node v_u are reserved to communicate in the time slot t_i using channel f_j. The route list is a set of nodes along the reserved path. For the nodes receiving the RREQ will take following steps.

(1) It checks its hop count distance to the destination d. If $H_G(v, d)$ > TTL, node v discards this request. This ensures that RREQ packet will be forwarded in the right direction towards destination.

(2) Otherwise it runs Algorithm 1 and Algorithm 2 to do one hop reservation.

(3) If one hop reservation by Algorithm 1 and Algorithm 2 is successful, node v updates TTL(TTL-1), the remaining latency bound RLb based on the result of one hop reservation, and appends v to the reservation list.

(4) Node v broadcasts the updated RREQ, and the two-hop neighbors of node v then updates $NB(t_i, f_j)$, which is an entry of the neighbor radio active table $NB(\mathrm{T}, \mathcal{F}) = \{NB(t_i, f_j) \mid \forall t_i\in \mathrm{T}, f_j\in \mathcal{F}\}$. $NB(t_i, f_j)$ is an integer to denote the number of interfering links of node v_i, which has announced its radio will broadcast or receive in time-slot t_i using channel f_i.

Besides $NB(\mathrm{T}, \mathcal{F})$, each node keep following two data structures.

(1) Self-radio broadcasting table: $SB(\mathrm{T}, \mathcal{F}) = \{SB(t_i, f_j) \mid \forall t_i\in \mathrm{T}, f_j\in \mathcal{F}\}$, where $SB(t_i, f_j)\in\{0, 1\}$ is an integer to denote whether an interface on v_i has announced it broadcasts in time-slot t_i using channel f_i.

(2) Self-radio receiving table: $SR(\mathrm{T}, \mathcal{F}) = \{SR(t_i, f_j) \mid \forall t_i\in \mathrm{T}, f_j\in \mathcal{F}\}$, where $SR(t_i, f_j)\in\{0, 1\}$ is similar to $SB(\mathrm{T}, \mathcal{F})$.

From the above described routing procedure, we can see that the step (2) is a local scheduling that minimizes one hop delay. Next, we will introduce the step in more detail.

3.2 One hop reservation

When a node v_j received the RREQ from its neighbor v_i, it first do an OR operation to compute its blocked table $BL(\mathcal{T},\mathcal{F})$ and find the possible available pairs of time-slot and channel for link e=(v_i, v_j). $BL_{vi,vj}=\{BL_{vi}(\mathrm{T}, \mathcal{F}) \cup BL_{vj}(\mathrm{T}, \mathcal{F})\}$, where $BL_{vi}(\mathrm{T}, \mathcal{F})=\{NB(t_i, f_j) \cup SB(t_i, f_j) \cup SR(t_i, f_j) \mid \forall t_i\in \mathrm{T}, f_j\in \mathcal{F}\}$ and BL_{vj}(T,

$\mathcal{F}$)={ $NB(t_i, f_j) \cup SB(t_i, f_j) \cup SR(t_i, f_j) \mid \forall t_i \in$ T, $f_j \in \mathcal{F}$ }. For example, the set of blocked time-slot and channel pairs are (t_2, f_1), (t_3, f_2), (t_4, f_1, f_2), and (t_5, f_2). Then, node v_j builds a two-dimension assignment status table, as shown in Table II. Node v_j uses this information to run reservation algorithm for this hop.

Following two types of delays are involved in the link reservation.

***(1) Scheduling Delay**:* counts from the last packet it received for this session until the last packet it sent out. This cannot be computed simply by subtracting the scheduled last receiving time-slot (*lastIn*) from the scheduled last sending-out time-slot (*lastOut'*). We should first perform a mapping on the *lastOut'*, then count the time elapse from the lastIn slot to the mapped last sending-out time-slot (*lastOut*), which is possibly different from *lastOut'*. After the mapping, this hop scheduling delay is *lastOut-lastIn*, if *lastOut* slot is within the same period as *lastIn* slot, otherwise is *lastOut-lastIn* + m, where m is the number of slots in a period.

For example, the previous hop has scheduled at time-slots t_1, t_4, t_5, and the current hop scheduled at time-slots t_2, t_3, t_6. Since the last sending-out time-slot (t_6) is after the last receiving time-slot, the scheduling delay is not 6-1=5. We should do a mapping first, $t_1 \rightarrow t_2$, $t_4 \rightarrow t_6$, $t_5 \rightarrow t_3$, and the scheduling delay is 3-5+m. However, we can improve the above mapping to shorten the scheduling delay, as follows, $t_1 \rightarrow t_3$, $t_4 \rightarrow t_6$, $t_5 \rightarrow t_2$, and the scheduling delay is reduced to 2-5+m. So the challenging question here is to find a distributed scheduling scheme, such that the overall delay of a route is minimized. Algorithm 1 shows a greedy method to schedule time slots with minimized delay.

Assuming that node v_i makes a reservation in the previous hop, that is T={(t_1, 3)(t_6, 2)}, and the available resources in the current link (v_i, v_j) is denoted as T'={(t_2', 1)(t_3',1) (t_4',2) (t_5', 1) (t_6', 2) (t_7', 2)} (See Table II), the scheduled slots is a set S={(t_2', 1)(t_3', 1) (t_4', 2) (t_5', 1)}and *Sche_delay*=3-6+7=4.

Algorithm 1 Slot Scheduling with Minimized Delay

Input: $T = \{(t_1, x_1)(t_2, x_2)\ldots(t_k, x_k)\}$, *RLb*, $T' = \{(t_1', x_1')(t_2', x_2')\ldots(t_d', x_d')\}$
Output: S
Variables: $h = 0$, *Sche_delay*
IF $\left(\sum_{i=1}^{k} x_i < \sum_{i=1}^{k} x_i\right)$ THEN
exit (FAIL)
ELSE
FOR EACH $t_i, \leftarrow 1\ldots k$ DO
WHILE ($x_i > 0$) do
IF there exists $t_j' > t_i$ then
$j = \arg\min \{t_j' \mid t_j' > t_i\}$
IF $x_j' > x_i$ THEN
$S = S \cup \{(t_j', x_i)\}$, $x_j' = x_j' - x_i$, $x_i = 0$
ELSE
$S = S \cup \{(t_j', x_j')\}$, $x_i = x_i - x_j'$, remove (t_j', x_j') from T'
END IF
ELSE
$j = \arg\min \{t_j'\}$
IF $x_j' > x_i$ THEN
$S = S \cup \{(t_j', x_i)\}$, $h = h + x_i$, $x_j' = x_j' - x_i$, $x_i = 0$
ELSE
$S = S \cup \{(t_j', x_j')\}$, $h = h + x_j'$, $x_i = x_i - x_j'$, remove (t_j', x_j') from T'
END IF
end if
end while
end for
end if
Sort the pairs (t_i'', x_i'') in S in ascending order by t_i''
IF $h > 0$ THEN
LastOut = t_q'', $q = \min\{j \mid \sum_{i=1}^{j} x_i \geqslant h\}$
ELSE
LastOut = t_q'', $q = \arg\max\{t_j''\}$
END IF
IF $t_k < t_q''$ THEN
Sche_delay $= t_q'' - t_k$
ELSE
Sche_delay $= t_q'' - t_k + m$
END IF
IF *RLb-Sche_delay*>0 THEN
return (SUCCESS);
ELSE
return (FAIL);
END IF

(2) Switching Delay: it is defined as the channel switching overhead incurred at each hop (v_i, v_j) for packet transmissions of a session in one schedule period. For example, if we assign f_1 to R_{i1} and R_{j1}, there is no channel switching and switching delay is 0. If we as-

Table II *Channel assignment status*

	t_1	t_2	t_3	t_4	t_5	t_6	t_7
R_{i1}	f_1		f_1		f_1		
R_{i2}	f_2	f_2					
R_{j1}	f_1		f_1		f_1		
R_{j2}	f_2	f_2					
R_{j3}							

Algorithm 2 Channel Assignment with Minimized Cost

Input: T = $\{(t_1, L_1, R_1), (t_2, L_2, R_2), \ldots, (t_b, L_b, R_b)\}$, *RLb*
Const: D_{cs}
Variables: *Switch_times*

```
FOR EACH t_i ← 1…b DO
    Construct a weighted bipartite graph G=(L_i, R_i);
    Find a minimal weighted matching M in bipartite graph from L_i to R_i using the K-M Algorithm;
    Switch_times ← Switch_times + w(M);
END FOR
IF RLb-Switch_times*D_cs > 0 THEN
    return (SUCCESS);
THEN
    return (FAIL);
END IF
```

sign f_2 to R_{i1} and R_{j1}, both radios incur 2 channel switches and the switching delay is (2 + 2)*switchOverhead. So the challenging question is to assign channels to radios to minimize the switching delay.

After determining scheduling delay by algorithm 1, a list of partial reservation for this hop is given by:

$\{(t_1, L_1, R_1), (t_2, L_2, R_2), \ldots, (t_b, L_b, R_b)\}$.

Referring to the previous example, after slot scheduling, the reservation can be presented as $\{(t_2, L_2, R_2), (t_3, L_3, R_3), (t_4, L_4, R_4), (t_5, L_5, R_5)\}$, where $L_2=\{f_1\}$, $L_3=\{f_2\}$, $L_4=\{f_1, f_2\}$, $L_5=\{f_2\}$, $R_2=\{R_{i1}, R_{j1}, R_{j3}\}$, $R_3=\{R_{i2}, R_{j2}, R_{j3}\}$, $R_4=\{R_{i1}, R_{i2}, R_{j1}, R_{j2}, R_{j3}\}$, $R_5=\{R_{i2}, R_{j2}, R_{j3}\}$. By considering each (t_i,L_i,R_i) tuple in order, the problem of assigning all channels in L_i to some pair of radios in R_i with minimal total switching delay can be converted to finding a perfect matching with minimal total weight in a bipartite graph from L_i to R_i using the Kuhn-Munkras (K-M) algorithm[18]. Algorithm 2 shows the detail of the channel assignment with Minimized Cost.

For the above example, the output of the channel assignment is $f_1 \xrightarrow{0} R_{i1}, f_2 \xrightarrow{0} R_{i2}$, $f_1 \xrightarrow{0} R_{j1}$, $f_2 \xrightarrow{0} R_{j2}$, and the switching delay is 0.

From the above description on one-hop reservation, we can see that two phases are involved in a local scheduling, namely slot scheduling and channel assignment. This two-phase local scheduling scheme is designed to minimize one-hop delay, as well as energy consumption.

IV. PERFORMANCE EVALUATION

The performance of our proposed protocol is evaluated by simulations. We put 225 nodes uniformly in a square area which is 200 meters by 200 meters. Each node is equipped with 2 radios, and there are 48 channels available in the network. The time cost of channel switching is set to 50 mini-second and the energy cost is 1940 micro-joule. The transmission range of each node is about 20 meters, and the interference range is about 40 meters. We specify the nodes located in the left-bottom corner as sources, while the node at the right-upper corner as the gateway. The packet length is 36 bytes, and the data rate of wireless channel is 76.8 Kbps. For each period of time, it consists of 7~21 time slot, and each slot is 4 milli-seconds.

For comparison, we implemented the R-MMAC in the simulator. R-MMAC is a multi-channel MAC protocol adapted from TMMAC[11]. The main difference between R-MMAC and DS-MMAC is in their way to schedule time slots and channels at each hop, which is the featured work of this paper. Following three metrics are used to evaluate the performance of DS-MMAC and R-MMAC.

1) Packet delivery ratio: the ratio of number of packets received by the gateway to the number of packet sent by the source nodes.

2) End-to-end delay: the time taken for a packet to be transmitted across a network from source to destination, which is an accumulation of the transmission delay, scheduling delay and switching delay at each hop.

3) Average energy consumed for each packet: the ratio of total energy consumed in the network to the number of packets delivered to the gateway.

4.1 Packet delivery ratio

Figure 3 shows the different packet delivery ratios of R-MMAC and DS-MMAC with increasing number flows. We can see that both protocols can achieve more than 95% of delivery ratio when the number of flows is less than 65, because they both employ TDMA to assign channels. However, with more number of sessions are launched in the network, the packet delivery of R-MMAC starts to decline by up to 5%, which may be incurred by some collisions due to its random assignment of channels at each hop. For DS-MMAC, it can keep packet delivery ratio at about 95% even with increasing number of flows. Hence, we can see that in scenarios where a number of sensing devices in the AAL system detect an event and simultaneously transmit the detected event, DS-MMAC can provide high reliability of packet delivery.

4.2 End-to-end delay

Since the DS-MMAC protocol employs a local scheduling algorithm that minimizes one hop delay when making link reservation at each hop, as shown in Figure 4, DS-MMAC has much lower end-to-end delay than that of R-MMAC, particularly decreased by up to 59%. Besides that, DS-MMAC can provide bounded end-to-end delay as required, since it considers the remaining delay when scheduling slots and assigning channels at each hop. Here, we set the delay bound of each packet to 500 ms, so with the number of flows increases, DS-MMAC can still keep the maximum end-to-end delay around 500 ms. For R-MMAC, because it schedules slots and assigns channels randomly at each hop, the delay can be accumulated and increasing rapidly, especially when the number of flows is more than 25, its delay increases linearly until to about 1200 ms. The reason why the delay stops increasing when the number of flow is larger than 65, is that packets will be dropped from this point (see Figure 3) and the network is saturated. Hence, we can see that even when a number of sensing devices in the AAL system detect an event and simultaneously transmit the detected event, DS-MMAC can ensure bounded low delay, which is the mainly expected performance of our protocol.

4.3 Average energy consumed for each packet

Figure 5 show the average energy consumed

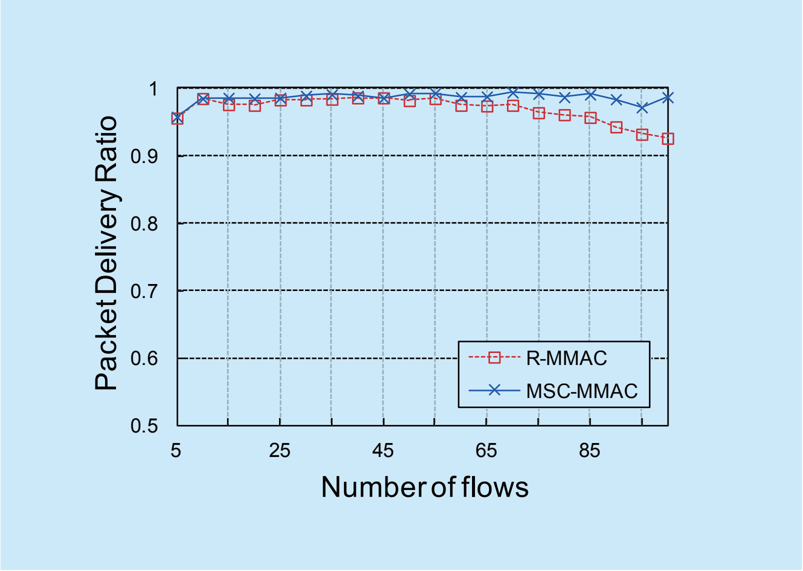

Fig.3 *Packet delivery ratio vs. number of flows*

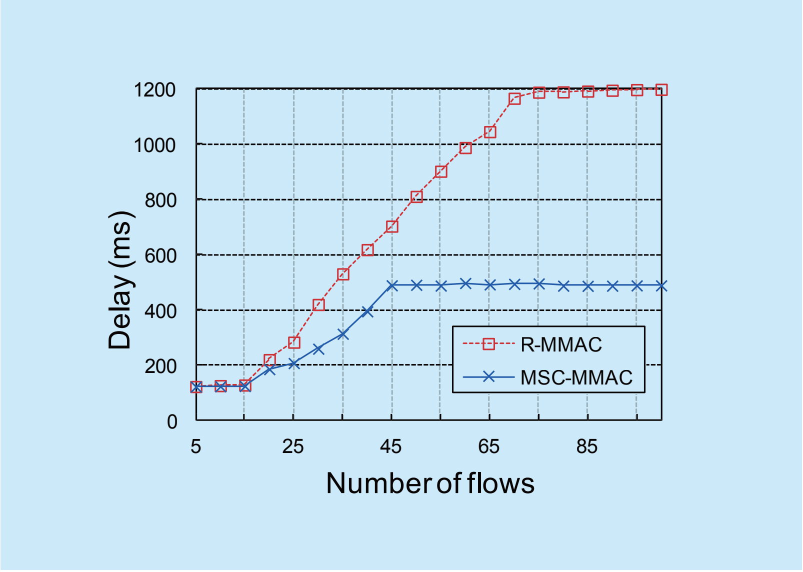

Fig.4 *End-to-end delay vs. number of flows*

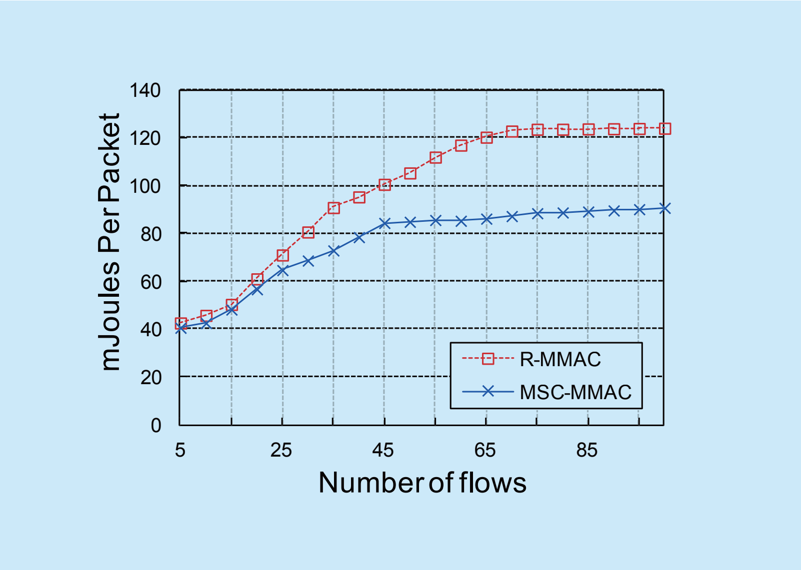

Fig.5 *Energy consumed for each packet vs. number of flows*

for each packet with different MAC protocols. We can see that with the increasing number of flows, for both DS-MMAC and R-MMAC, more energy is needed to deliver a packet from the source to the gateway. However, the energy consumption for DS-MMAC rises more slowly than R-MMAC with increasing number of flows. Ultimately, DS-MMAC can reduce the energy consumption by about 28% compared with R-MMAC. This is mainly due to the reduced switching times by employing the one-hop scheduling at each hop in DS-MMAC. Hence, we can see that by taking switching cost into account, DS-MMAC can save energy and prolong lifetime of the embedded sensing devices in AAL system.

V. Conclusion

In AAL applications, occurrence of a concerned event will trigger the embedded devices to collect data and send it to the gateway node. In such a scenario, the collected data are required to be received by the gateway node promptly and to be forwarded to the remote server reliably. It is difficult for the traditional low-power embedded network (e.g. wireless sensor network), which is composed of nodes with one radio module and sharing a wireless channel, to achieve this goal. With development of embedded systems, each sensor node or embedded sensing device is equipped with more than one radio modules, and more and more researchers exploit the underlying support of multi-channel communication to fundamentally solve the problem. However, most of those protocols ignore the cost of channel switching, which can have reverse effect on network performance, especially latency of data delivery.

In this paper, we propose a delay-sensitive multi-channel MAC protocol, named DS-MMAC, which minimizes delay of data transmission in the embedded network of AAL applications. The novelty of the protocol is that an efficient distributed time slot scheduling and channel assignment algorithm is combined with the process of route establishment, which ensure that the reserved links satisfy the required bandwidth and with minimized delay.

The performance of the proposed protocol is evaluated through extensive simulations. Results show that DS-MMAC outperforms R-MMAC in terms of end-to-end delay, as well as packet delivery ratio and energy usage for each packet. So for AAL applications, when a concerned event occurs, which will trigger the embedded devices to collect data and send it to the gateway node, DS-MMAC can bound low latency for delivering detected events in AAL system to the gateway, while providing high delivery reliability and low energy consumption.

Acknowledgements

The early work of DS-MMAC was done with cooperation with Dr. Hu and Prof. Xiangyang Li, and has been partially published in IEEE GlobeCom 2007 [19]. This work is supported by the International S&T Cooperation Program of China (ISTCP) under Grant No. 2013DFA10690 and the National Science Foundation of China (NSFC) under Grant No. 61100180.

References

[1] GARCIA N. M and RODRIGUES J.P.C. Ambitent

Assitant Living [M].USA: CRC Press, 2015: 7-40.
[2] BOTIA J.A, VILLA A, and PALMA J. Ambient Assisted Living System for in-home monitoring of healthy Independent Elders[J]. Expert Systmes with Applications, 39(9): 8136-8148, 2012.
[3] FAGERBERG G, KUNG A, WICHERT R, et al. Platforms for AAL Applications [C]// Proc. of the European Conference on Smart Sensing and Context (EuroSCC'10), Heidelberg: Springer-Verlag Berlin, 2010:177-201.
[4] YE W., HEIDEMANN J., and ESTRIN D. An Energy-Efficient MAC protocol for Wireless Sensor Networks [C]// Proc. of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'02), San Fracisco, CA: IEEE Press, 2002: 1567 -1576
[5] Dᴀᴍ T. van and LANGENDOEN K. An Adaptive Energy-efficient MAC Protocol for Wireless Sensor Networks [C]// Proc. of the 1st ACM Conference on Embedded Networked Sensor Systems (SenSys'03), Los Angeles, CA: ACM Press, 2003: 171–180.
[6] POLASTRE J, HILL J, and CULLER D. Versatile Low Power Media Access for Wireless Sensor Networks [C]// Proc. of the 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys'04), Baltimore, MD: ACM Press, 2004: 95–107.
[7] BUETTNER M, YEE G.V, ANDERSON E, and HAN R. X-MAC: A Short Preamble MAC protocol for Duty-cycled Wireless Sensor Networks [C]//Proc. of the 4th ACM Conference on Embedded Networked Sensor Systems (SenSys'06), Boulder, Colorado: ACM Press, 2006:307–320.
[8] SUN Y, GUREWITZ O, and JOHNSON D.B. RI-MAC: A Receiver Initiated Asynchronous Duty Cycle MAC Protocol for Dynamic Traffic Load [C]. // Proc. of the 6th ACM Conference on Embedded Networked Sensor Systems (SenSys'08), Raleigh, North Carolina, USA, Nov. 2008:1-14.
[9] SUN Y, GUREWITZ O, and JOHNSON D.B. DW-MAC: A Low Latency, Energy Efficient Demand-Wakeup MAC Protocol for Wireless Sensor Networks [C]. //Proc. of the 9th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc'08), Hong Kong, China, 2008: 53–62.
[10] ZHOU G, HUANG C, YAN T. HE T, STANKOVIC J. A, and ABDELZAHER T. F. MMSN: Multi-Frequency Media Access Control for Wireless Sensor Networks [C]// Proc. of the 25th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'06), Barcelona, Spain: IEEE Press, 2006: 1–13.
[11] ZHANG J, ZHOU G, HUANG C, SON S.H, and STANKOVIC J.A. TMMAC: An Energy Efficient Multi-Channel MAC Protocol for Ad Hoc Networks [C]// Proc. of the IEEE International Conference on Communications (ICC'07), San Fracisco, CA: IEEE Press, 2007: 3554–3561.
[12] KIM Y, SHIN H, and CHA H. Y-MAC: An Energy-Efficient Multi-channel MAC Protocol for Dense Wireless Sensor Networks [C]// Proc. of the 7th International Conference on Information Processing in Sensor Networks (IPSN'08), St. Louis, Missouri: ACM Press, 2008: 53–63.
[13] WU Y, STANKOVIC J.A, HE T, and LIN S. Realistic and Efficient Multi-Channel Communications in Wireless Sensor Networks [C]// Proc. of the 27th IEEE Conference on Computer Communications (INFOCOM'08), San Fracisco, CA: IEEE Press, 2008: 1193–1201.
[14] RAMAN B, CHEBROLU K, BIJWE S, et al. PIP: A Connection-Oriented, Multi-Hop, Multi-Channel TDMA-based MAC for High Throughput Bulk Transfer[C] //Proc. of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys'10), Zurich, Switzerland: ACM Press, 2010: 15-28
[15] CHEN Haiming, CUI Li and LU Shilong. An Experimental Study of The Multiple Channels and Channel Switching in Wireless Sensor Networks [C]// Proc. the 4th International Symposium on Innovations and Real-time Applications of Distributed Sensor Networks (IRADSN'09), Hangzhou, China: Conference Press, 2009: 1-8.
[16] TI. CC100 Single-Chip Very Low Power RF Transceiver [EB/OL]. http://www.ti.com/lit/ds/symlink/cc1000.pdf, 2009.
[17] ANSARI J, ZHANG X, and MAHONEN P. Demo Abstract: Multi-radio Medium Access Control Protocol for Wireless Sensor Networks [C]// Proc. of the 5th ACM Conference on Embedded Networked Sensor Systems (SenSys'07), Sydney, Australia: ACM Press, 2007: 403–404
[18] GUPTA A. and YING L. On Algorithms for Finding Maximum Matchings in Bipartite Graphs [R]. USA: IBM T. J. Watson Research Center, 1999.
[19] HU Yi, LI Xiang-Yang, CHEN Hai-Ming, and JIA Xiao-Hua. Distributed call admission protocol for multi-channel multi-radio wireless networks[C] //Proc. of the IEEE Global Telecommunications Conference (GlobeCom'07), San Fracisco, CA: IEEE Press, 2007: 2509 - 2513

Biographies

Haiming Chen, Ph.D, is now an assistant professor at the Institute of Computing Technology, Chinese Academy of Sciences. His current research areas include wireless, ad hoc, sensor networks and networked embedded computing systems. He is a member of IEEE and ACM. Email: Chenhaiming@ict.ac.cn

Li Cui, Ph.D, is currently a Professor and the leader of the wireless sensor network laboratory at the Institute of Computing Technology, Chinese Academy of Sciences. Her current research is focused on sensor technology and wireless sensor networks. She is a fellow of IET and a member of IEEE. Email: lcui@ict.ac.cn