Sustainability **2014**, *6*, 4326-4338; doi:10.3390/su6074326

OPEN ACCESS

sustainability

ISSN 2071-1050

www.mdpi.com/journal/sustainability

Article

Bitmap-Wise Wireless M-Bus Coordination for Sustainable Real Time Energy Management

Kwang-il Hwang * and Sung-wook Nam

Department of Embedded Systems Engineering, Incheon National University Incheon, Incheon 402-772, Korea; E-Mail: nam1260@nate.com

* Author to whom correspondence should be addressed; E-Mail: hkwangil@incheon.ac.kr; Tel.: +82-32-835-8762; Fax: +82-32-835-0782.

Received: 5 June 2014; in revised form: 24 June 2014 / Accepted: 1 July 2014 / Published: 14 July 2014

Abstract: Even though WM-Bus is being considered to be the most promising network protocol for smart metering, it is not suitable for a sustainable real-time home energy management system (HEMS), which requires higher reliability and longer lifetime despite real time bi-directional communications. Therefore, in this paper we propose a Bitmap-wise WM-Bus (BWM-Bus), coping well with sustainable real-time HEMS. In particular, the proposed scheme addresses the several problems in WM-Bus for HEMS by introducing novel functions: asynchronous meter trigger, adaptive slot scheduling, and bitmap-wise retransmission request. Through experiments, we demonstrate that BWM-Bus guarantees higher data success ratio with lower data aggregation time, as well as longer lifetime than WM-Bus standard.

Keywords: home energy management systems; bitmap-wise network coordination; smart metering

1. Introduction

Home Energy Management System (HEMS) [1] is an emerging technology capable of maximizing energy efficiency, as well as saving energy, by controlling and managing a variety of energy, which are produced and consumed in a home, via Home Area Network (HAN) associated with Wireless AMR (Automated Meter Reading). In particular, WM (Wireless Metering)-Bus [2,3], as a metering network minimizing OSI layer, is recently being regarded as one of the most promising smart metering networks. Therefore, a great deal of research on WM-Bus has been conducted [4–11]. These include research for improving performance, as well as various applications using WM-Bus. However, WM-Bus standard is optimized for simple metering service, in which meter data is collected one time per day or a few times per month, only for billing, so that as such, WM-Bus is not suitable for HEMS, which requires higher reliability and longer lifetime despite real time bi-directional communications.

Therefore, in this paper we propose an enhanced WM-Bus, which provides real-time data aggregation, based on high data success ratio, despite more energy efficiency. The enhanced WM-Bus is capable of maximizing ultra low-power functionality, utilizing preamble-based low-power listening, and performing dynamic scheduling and efficient retransmission via a novel bitmap-wise network coordination.

2. Related Work

In recent, a great deal of research on WM-Bus is being actively conducted. Rorato *et al*. [4] proposed a low-cost smart metering system based on WM-Bus. The proposed network architecture is based on star topology, consisting of access point and meters, and the authors designed low-power firmware to prolong the lifetime over 10 years. Craemer *et al*. [5] investigated state-of-the-art smart metering standards and emphasized the importance of WM-Bus in smart metering. In addition, there has been some research focusing on the performance evaluations of WM-Bus. Spinsante *et al*. [6] performed performance evaluations of WM-Bus in terms of PER (Packet Error rate) test and maximum transmission coverage. Kuder *et al*. [7] implemented a WM-Bus simulator using ns-3 network simulator. Gubisch *et al*. [8] emphasized necessity of a proper trade-off among different modes in WM-Bus. Some research also focused on the performance enhancement of WM-Bus standard. Jacobsen *et al*. [9] proposed a data recovery method, using side information to solve data errors, which can be caused in WM-Bus S-mode. Sikora *et al*. [10] designed a portable and flexible metering protocol stack for gas metering application using W-MBus. Squartini *et al*. [11] proposed a W-MBus-based sensor node design in smart water grid. Liu *et al*. [12] proposed an energy efficient mesh network to be considered for large scale smart metering network. Sinha *et al*. [13] evaluated the performance of data aggregation protocols for sensor networks. In addition, Yoon *et al*. [14] also proposed an energy efficient protocol, using success rate in wireless networks.

However, since the existing research has been focused on the architecture and performance of WM-Bus only for general smart metering networks, a new approach to WM-Bus protocol optimized for real time HEMS is necessary. Therefore, in this paper we first point out some problems of WM-Bus for real-time HEMS, and then propose an enhanced WM-Bus.

3. Experimental Study of WM-Bus for HEMS

In this Section, experimental study of the WM-BUS protocol on HEMS test-bed is presented. In particular, we evaluate how sustainable WM-BUS is in real-time HEMS, by analyzing performance with respect to data success ratio and data aggregation time.

3.1. Experimental Environment

To evaluate the performance of WM-BUS, we implemented a WM-BUS test-bed as shown in Figure 1. The HEMS test-bed is composed of 1 collector, which plays a role in energy service portal, and maximum 10-meter devices. For a WM-Bus standard protocol, we experimented R2 mode, which can provide flexible data gathering and energy efficiency, and CSMA-based LBT is implemented according to guideline presented in WM-Bus user guide [15], and the detailed operation is illustrated in Figure 2.

Figure 1. WM-Bus based HEMS Test bed.

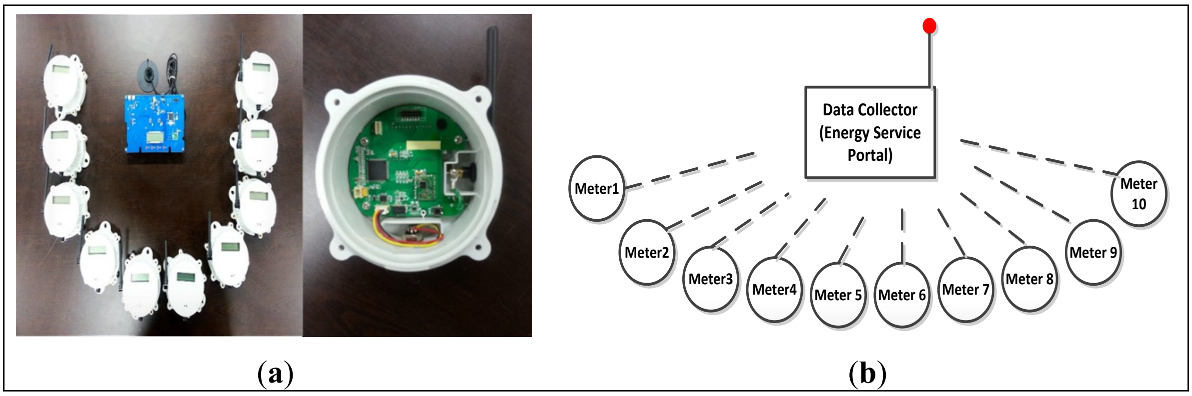
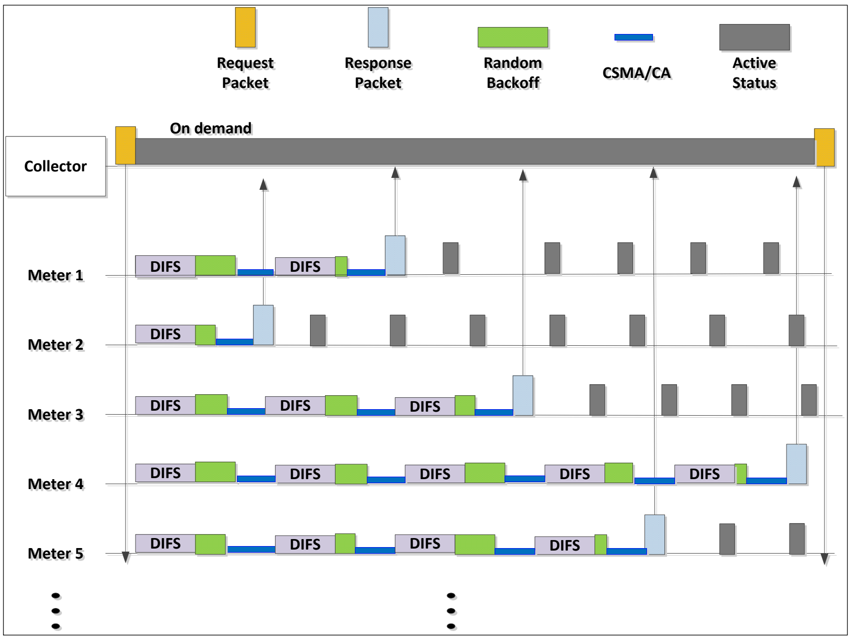


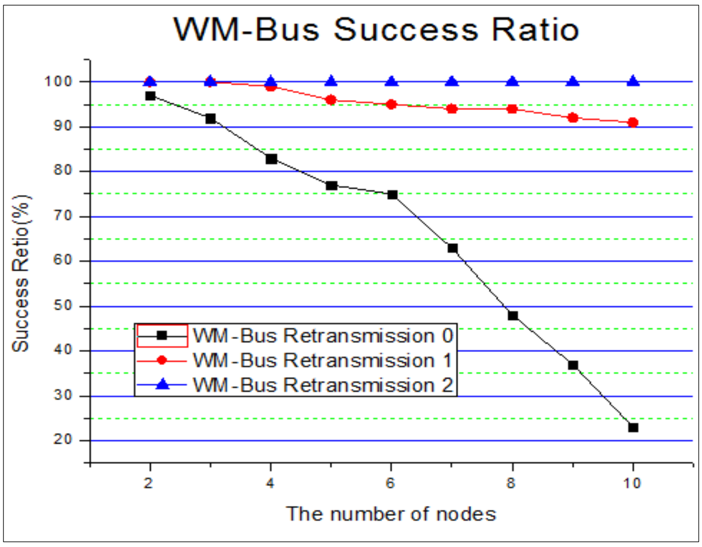
Figure 2. WM-Bus R2 mode communication flow.



3.2. Success Ratio

First, we observed data success ratio with respect to varying the number of meter devices from 2 to 10. As shown in Figure 3, experimental result shows that success ratio is rapidly declined as the number of nodes increases. The performance is enhanced by additional retransmission requests of more than 2 times. However, continuous retransmissions might cause longer data aggregation time and inconsistent delay distribution due to random back-off effect.

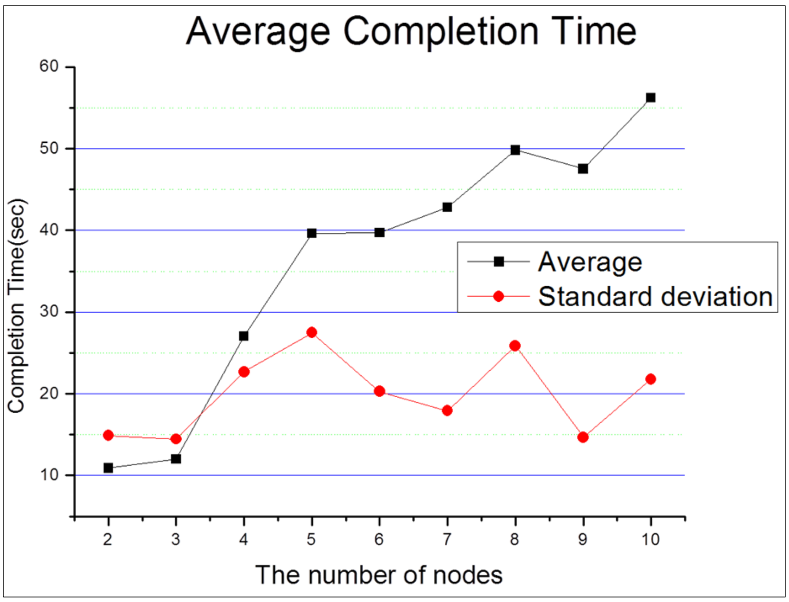
Figure 3. LBT-based WM-Bus data success ratio.



3.3. Average Completion Time

In addition to success ratio, we also observed average data completion time of each node. Figure 4 shows the result of average completion time and standard deviation with respect to varying the number of nodes. Experiments are repeated 100 times and the result represents its average value. The result first reveals that the standard deviation is rarely affected by the number of nodes, but also shows that average completion time is significantly increased as the number of nodes increase. Therefore, the result demonstrates that WM-BUS shows inconsistent data aggregation time due to random back-off, and delay deviation is fluctuated due to addition retransmissions caused by transmission failures.

Figure 4. Average completion time.



3.4. Summary

Through various experiments, we verified that the pure WM-BUS is not suitable for real-time HEMS environment. In particular, even though data failure caused by collisions can be reduced by applying retransmissions, since LBT might require additional retransmissions as the number of nodes increases, data completion time is nondeterministic and the possibility of failures still remains unsolved. Therefore, for a sustainable real-time HEMS, WM-BUS should be enhanced so as to ensure a higher success ratio and deterministic data aggregation time.

Figure 5. Bitmap libraries designed for BWM-Bus.

```
<Variables>
 A[]: BITMAP ARRAY
 TA[]: Temporary BITMAP ARRAY
 IA[]: Inverted BITMAP ARRAY
 sA: Size of bitmap data type
  ( eg., unsigned char A[BITMAP_ARRAY_SIZE],
    in that case we have sA = 8 )
 num: A value for making bitmap
 ID: identifier of node
 index: indicator of array index
 position: bit position in a A[x],
  where x = 0, 1, 2, ... BITMAP_ARRAY_SIZE

**Init_bitmap(A,BITMAP_ARRAY_SIZE)***************

memset(A,0,BITMAP_ARRAY_SIZE)

************************************************

** Insert_to_bitmap (A, ID)************************

init_bitmap (TA, MAX_BITMAP_ARRAY )
memcpy (TA,A,MAX_BITMAP_ARRAY)
num ← ID – 1
index ← num / sA
position ← num mod sA
A[index] ← A[index] | 0x01 << position

************************************************

**Is_set_in_Bitmap (A,ID)***********************

num ← ID – 1
index ← num / sA
position <- num mod sA
if(A[index] & (0x01 << position)
     return 1;
else
     return 0;

************************************************

**Num_of_setbits_in_bitmap (A)*****************

seq ← 0
size ← sizeof(A)
for i ← 0 to size
     for j ← 0 to sA
               if (A[i] & (0x01 << j ))
                         seq ← seq + 1
               j ← j + 1
     i ← i +1
return seq

************************************************
```

```
**Waiting_seq_in_bitmap (A, ID) *****************

seq ← 0
num ← ID -1
index ← num / sA
position ← num mod sA
for i ← 0 to index
     for j ← 0 to 8
               if (A[i] & (0x01 << j))
                         seq ← seq +1
               j ← j + 1
     i ← i + 1
for j ← 0 to position
     if (A[i] & (0X01 << j))
               seq ← seq + 1
     j ← j + 1
return seq – 1

************************************************

**Remaining_seq_in_bitmap (A, ID)****************

seq ← 0
num ← ID -1
index ← num / sA
position ← num mod sA
for j ← 0 to position
     if (A[i] & (0x01 << j))
               seq ← seq +1
     j ← j + 1
for i ← index + 1 to MAX_BITMAP_ARRAY
     for j ← 0 to sA
               if (A[i] & (0X01 << j))
                         seq ← seq + 1
               j ← j + 1
     i ← i + 1
return seq

************************************************

** Invert_bitmap (A,IA,SIZE)************************

num ← SIZE -1
index ← num / sA
position ← num mod sA
for i ← 0 to index + 1
     if (i == index)
               for j ← 0 to position + 1
                         if(! (A[i] & (0x01 << j )))
                                   IA[i] ← IA[i] | (0x01 << j)
                         j ← j + 1
     else
               for j ← 0 to sA
                         if(! (A[i] & (0x01 << j)))
                                   IA[i] ← IA[i] | (0x01 <<j)
                         j ← j + 1

     i ←i + 1

************************************************
```

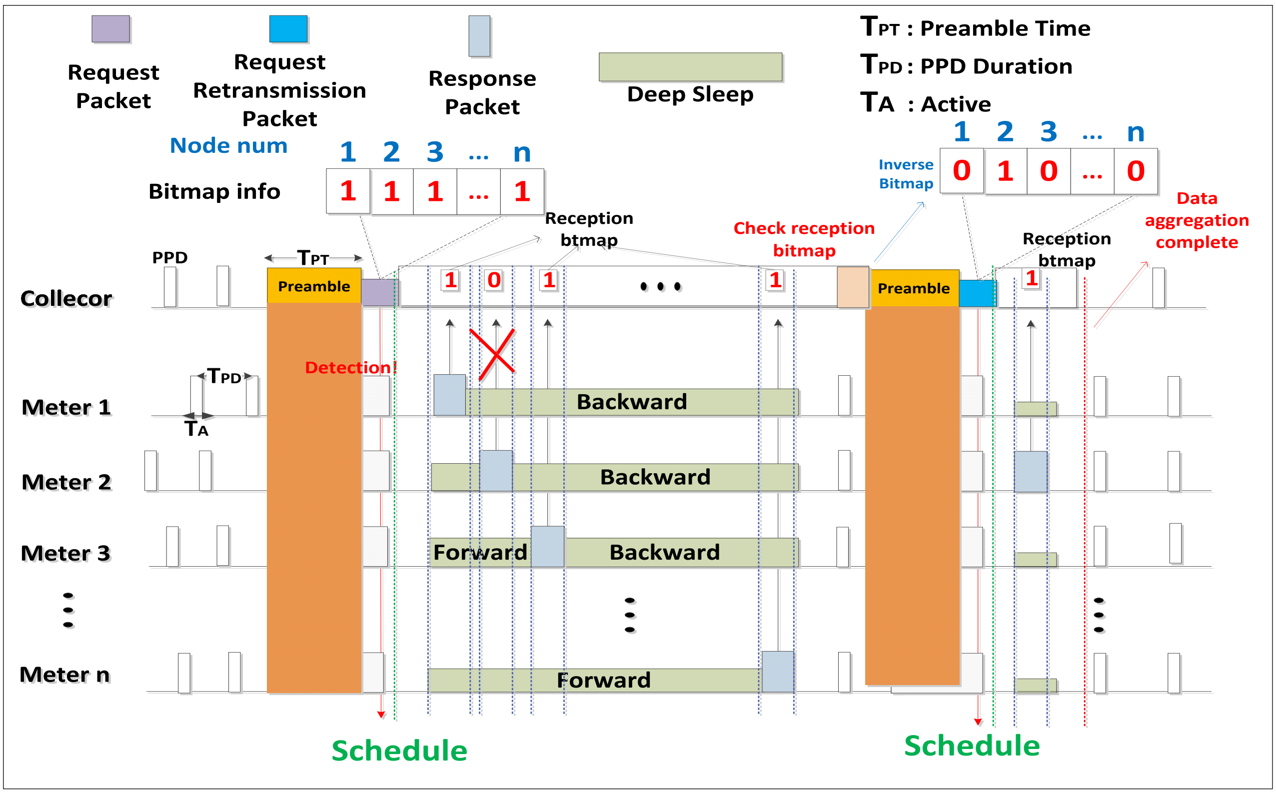
4. Bitmap-Wise WM-Bus Coordination

As mentioned in the previous Section, WM-Bus basically utilizes Listen Before Talk (LBT) based on random back-off for multiple access. Even though LBT is simple and adaptive for a small-scale network, it might not be suitable for HEMS due to increasing collision probability and inconsistent data aggregation time. Therefore, in order to solve the limitation of LBT-based WM-Bus, we propose bitmap-wise network coordination, associated with preamble-based low-power listening (LPL). Unlike LBT-based WM-Bus, depending on random back-off, the proposed scheme includes novel functionalities capable of asynchronously triggering each node, performing periodic preamble sensing (PPS) to detect the preamble transmission of a sender, and supporting various data transmission models, such as broadcasting, unicsting, multicasting, *etc*., by utilizing a bitmap-wise adaptive slot scheduling and retransmission request. In particular, one of the most specific features of the proposed scheme is based on bitmap utilization, thus that the proposed scheme is called BWM-Bus, and we developed essential bitmap libraries. The detailed pseudo code of each bitmap function is presented in Figure 5.

4.1. Asynchronous Meter Trigger

As shown in Figure 6, BWM-Bus devices repeat PPS in normal state, so it is possible to keep minimum active time within a period by detecting, not a packet, but a preamble transmission. Therefore, BWM-Bus devices can save more energy by staying asleep for longer time in the same period, compared to WM-Bus, which should wait for entire packet reception for the active period to periodically check a packet transmission.

Figure 6. An operational example of BWM-Bus.

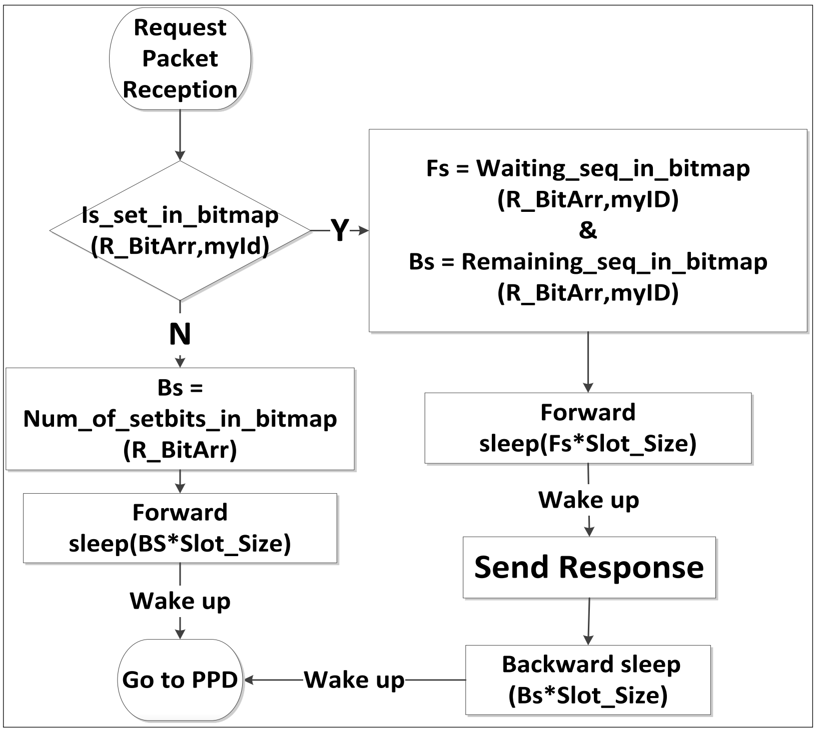


Moreover, a collector first transmits preamble for T_{ps} to trigger devices performing PPS, and then transmits a request packet. In particular, a collector can request broadcasting, unicasting, and multicasting to devices by sending a request packet containing a bitmap representing addresses of target devices. Each device receives a request packet from the collector, and performs further processing after checking whether its ID is set in the bit or not.

4.2. Bitmap-Wise Adaptive Slot Scheduling

One of the outstanding features of BWM-Bus can solve several problems occurring in LBT-based WM-Bus, such as data loss and inconsistent aggregation time, by adaptively scheduling devices with bitmap information contained in a request packet from a collector. Figure 7 shows a bitmap-wise adaptive slot scheduling process in a meter device. First, upon reception of a request packet, a device (meter) calculates its slot number to be assigned and estimated total aggregation time through its ID sequences from the received bitmap (R_BitArr). In addition, BWM-Bus can use minimum energy for its data transmission by staying in deep sleep for both of the forward and backward time of its own data transmission slot. A deep sleep state can save much more energy by maintaining a power down state of all ICs in a device for the scheduled duration. Each of forward sleep and backward sleep times can be also obtained by its ID sequences from the received bitmap. After completion of a backward deep sleep schedule for the request, each meter repeats PPS again.

Figure 7. Bitmap-wise adaptive scheduling flowchart.

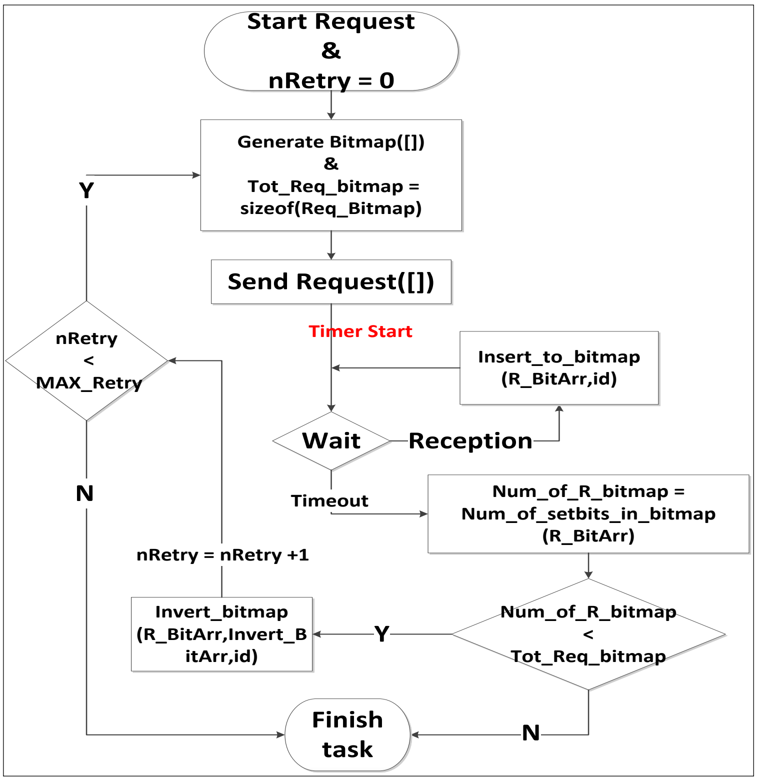


4.3. BRQ: Bitmap-Wise Retransmission Request

For WM-Bus, the response packet of each meter is acknowledged individually by the collector, and this results in more increasing collision possibility as the number of meters increases. Eventually, retransmission requests are also increased. In order to solve the increasing retransmissions by individual ACK, BWM-Bus utilizes Bitmap-wise Retransmission reQuest (BRQ). Since the BRQ does not allow individual ACK from a collector, collision probability by individual ACK can be significantly decreased, and the retransmission algorithm complexity is also reduced by allowing identical processing of the retransmission and general request.

Figure 8 shows BRQ processing in a collector. Collector first generates a bitmap representing target devices and sends a request containing the request bitmap (Req_bitmap). Since the collector can estimate total data aggregation completion time based on the generated bitmap, it starts a timer and waits for responses from meters. During data aggregation phase, an empty bitmap, reception bitmap, is used to check a responder ID whenever response is received. As soon as the timer expires, the collector compares an original request bitmap and reception bitmap. If there is difference, as it means some nodes failed to transmit a response, retransmission for the failure nodes should be requested. It is of note that the retransmission request of BWM-Bus can be easily accomplished. For the retransmission, the reception bitmap is inverted, and then the request packet containing the inverted bitmap is transmitted. Figure 8 shows an example of BRQ associated with bitmap-wise adaptive slot scheduling.

Figure 8. Bitmap-wise retransmission request flowchart.



5. Performance Evaluations

This Section evaluates the comparative performance of BWM-Bus with WM-Bus standard on the developed HEMS test-bed. By evaluating successful data transmission ratio and average data aggregation, we verify if the proposed BWM-Bus is suitable for real-time HEMS. In addition, we investigate how energy-efficient the BWM-Bus is by analyzing the device lifetime.

5.1. Experimental Environment

On the same test-bed, as described in Section 3, WM-Bus protocol and BWM-Bus protocol are implemented respectively. In addition, Table 1 presents key parameters used in the experiments.

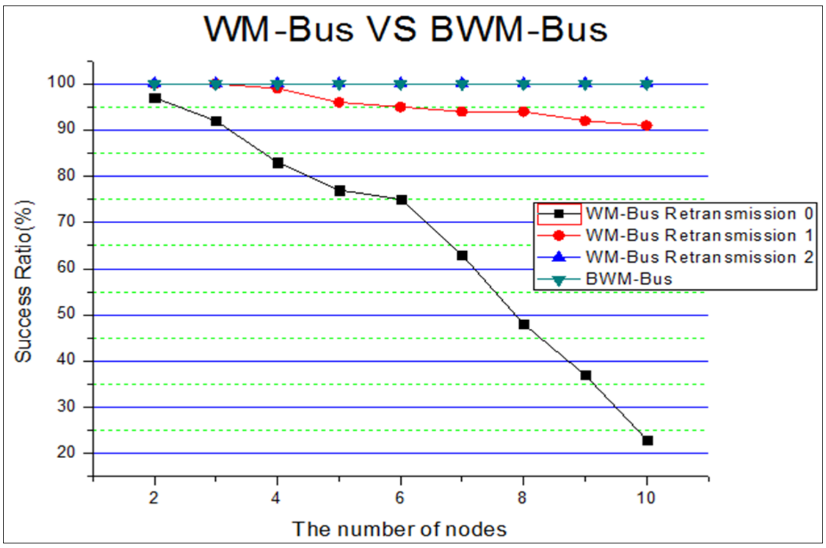
Table 1. Key Parameters used in experiments.

Parameter	Description	Value
TA	Active time in PPD	15.6 ms
TS	Sleep time in PPD	1000 ms
TPT	Preamble time for request	1045.6 ms
TDT	Data transmission time	1000 ms
I(a)	Active power consumption	22 mA
I(s)	Sleep power consumption	0.033 mA
BAT	Battery capacity	24,000 mAh
R_Backoff	Random back-off time	$2^n - 1 * \text{BaseSlotDuration}$ $(4 \leq N \leq 8)$
BaseSlotDuration	A unit slot duration for random back-off	16 ms

5.2. Success Ratio

First, we observed data success ratio with respect to the request of collector, according to varying the number of meter devices (2–10). As shown in Figure 9, WM-Bus with no retransmission presents rapid degradation of success ratio as the number of nodes increases, but through retransmissions the success ratio is improved, and eventually 100% success ratio is achieved at two retransmissions. However, in the case that retransmission requests increase, data aggregation time gets longer and is inconsistent due to random back-off effect. On the other hand, the result also shows that BWM-Bus guarantees 100% success ratio even for 10-meter devices without any retransmission. This is because BWM-Bus is capable of minimizing collision probability among packets by adaptively scheduling meter devices that are asynchronously triggered by preamble of a collector, as if devices use TDMA.

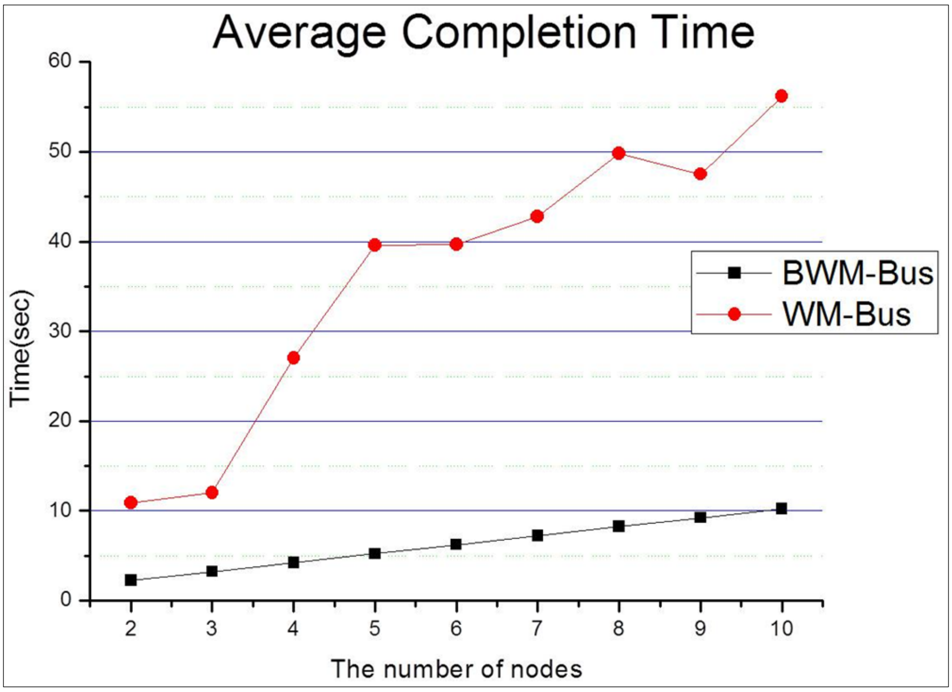
Figure 9. Success ratio *vs*. the number of nodes.



5.3. Average Response Completion Time

The second observation is average response completion time. As shown in Figure 10, WM-Bus shows inconsistent response completion time, and moreover the completion time is considerably increased as the number of devices increases. This results from random back-off delay and additional retransmission occurring due to increasing collision probability as presented in Subsection 4.2. On the other hand, BWM-Bus shows a smooth linear increment as the number of devices increases. This is because BWM-Bus can minimize collision possibility by assigning a mutually exclusive slot through bitmap-wise adaptive slot scheduling and utilizing BRQ instead of individual ACK, and, thus, it guarantees smooth data aggregation time as the number of devices increases.

Figure 10. Average response completion time.

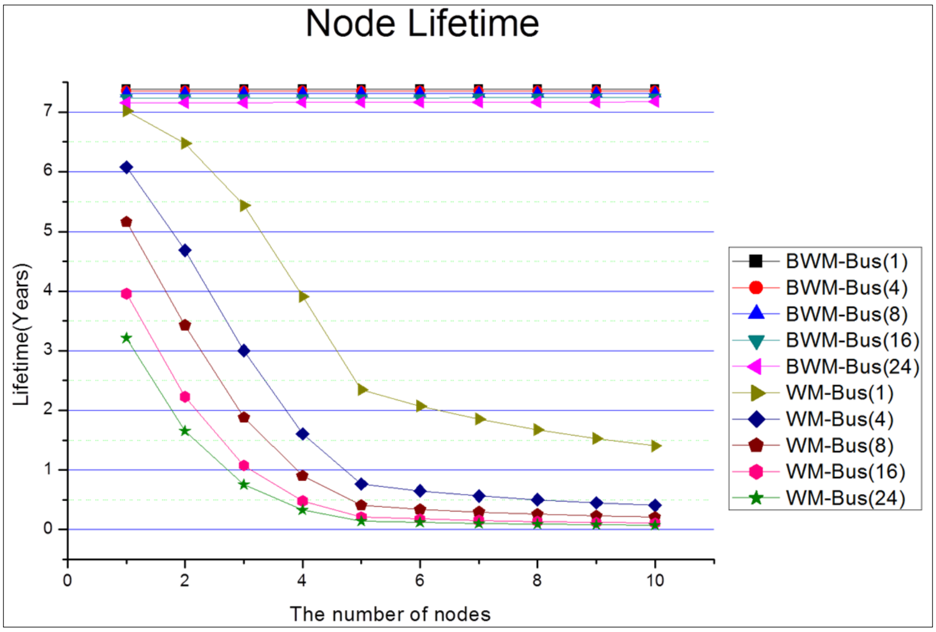


5.4. Meter Device Lifetime

One of the important design factors in WM-Bus is lifetime of a meter device. In particular, unlike general AMR or smart metering in which data gathering interval is almost fixed at one time or a few times per day, since HEMS requires real-time meter data gathering, the required energy consumption is more increased and thus lifetime of each meter device will be shortened. Therefore, we analyzed device lifetime with respect to varying data request rates, 1–24, for 24 h. For the analysis, we measured active duration and sleep duration, and obtained the estimated lifetime by applying current consumption in an active state and sleep state, respectively.

Figure 11 shows a comparative lifetime of WM-Bus and BWM-Bus. As the number of nodes increases, WM-Bus shows a rapid lifetime decrease. This results from increased energy consumption, as active duration is longer due to increasing collisions and retransmissions. On the other hand, BWM-Bus shows a longer lifetime than WM-Bus. Furthermore, it shows almost the same lifetime without regard to the number of devices. This is because BWM-Bus devices can remain in deep sleep for almost identical forward and backward sleep times by bitmap-wise adaptive scheduling, as with TDMA works. In addition, the BWM-Bus is designed based on preamble-based low power listening, so that, in spite of high request rate, a longer lifetime can be guaranteed.

Figure 11. Energy consumption.



6. Conclusions

Even though WM-Bus is being considered as the most promising network protocol for smart metering, it is not suitable for real-time home energy management systems (HEMS) due to increasing collision probability by random back-off and retransmissions. Therefore, we proposed a Bitmap-wise WM-Bus (BWM-Bus), to solve the problems in WM-Bus for HEMS. In particular, novel features of BWM-Bus, including asynchronous meter trigger, adaptive slot scheduling, and bitmap-wise retransmission request, make it possible for WM-Bus devices to satisfy real-time HEMS requirements.

In addition, through experiments, we demonstrate that BWM-Bus guarantees a higher data success ratio with a lower data aggregation time, as well as a longer lifetime than the WM-Bus standard. Finally, it is expected that the BWM-Bus will be able to contribute to extending the development of new WM-BUS applications

Acknowledgments

This work was supported by the Incheon National University Research Grant in 2014.

Author Contributions

The authors listed both contributed to the development of the idea and experiments contained within this paper, building on previous works as noted within the text. The paper's structure, case-study and revisions were led by the first author.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. ZigBee Smart Energy V2.0 Document, ZigBee Alliance, 2013. Available online: http://www.zigbee.org/Standards/ZigBeeSmartEnergy/ZigBeeSmartEnergy20Standard.aspx (accessed on 1 July 2014).
2. Communication systems for meters and remote reading of meters—Part 1: Data exchange; EN 13757-1:2002, Part 2: Physical and link layer; EN 13757-2:2004, Part 3: Dedicated application layer; EN 13757-3:2004, Part 4: Wireless meter readout (Radio meter reading for operation in SRD bands); prEN 13757-4:2011, Part 5: Relaying; EN 13757-1:2007. Available online: http://oms-group.org/en/standard-sources/ (accessed on 1 July 2014).
3. Steinbeis transfer center for embedded design and networking. Available online: http://www.stzedn.de (accessed on 15 December 2011).
4. Rorato, O.; Bertoldo, S.; Lucianaz, C.; Allegretti, M.; Notarpietro, R. An Ad-Hoc Low Cost Wireless Sensor Network for Smart Gas Metering. *Wirel. Sens. Netw.* **2013**, doi:10.4236/wsn.2013.53008.
5. De Craemer, K.; Deconinck, G. Analysis of State-of-the-art Smart Metering Communication Standards. In Proceedings of the IEEE Benelux Young Researchers Symposium 2010 in Electrical Power Engineering, Leuven, Belgium, 29–30 March 2010.
6. Spinsante, S.; Pizzichini, M.; Mencarelli, M.; Squartini, S.; Gambi, E. Evaluation of the Wireless M-Bus Standard for Future Smart Water Grids. In Proceedings of the 9th International on Communications and Mobile Computing Conference (IWCMC), Sardinia, Italy, 1–5 July 2013.
7. Kuder, Z.; Jacobsen, R.M. Feasibility of Wireless M-Bus Protocol Simulation. *Elektrorevue* **2012**, *3*, 1–5.
8. Gubisch, T.; Sikora, A. New Developments for Wireless M-Bus. In Proceedings of the Embedded Word Conference 2009, Nuremberg, Germany, 24–26 February 2009.
9. Melchior Jacobsen, R.; Popovski, P. Data Recovery using Side Information from the Wireless M-Bus Protocol. In Proceedings of the IEEE Global Conference on Signal and Information Processing, Texas, USA, 3–5 December 2013.
10. Sikora, A. Portable and Flexible Communication Protocol Stacks for Smart Metering Projects. *J. Electron. Sci. Technol.* **2013**, *11*, pp. 58–65.
11. Squartini, S.; Gabrielli, L.; Mencarelli, M.; Pizzichinini, M.; Spinsante, S.; Piazza, F. Wireless M-Bus Sensor nodes in Smart Water Grids: The Energy Issue. In Proceedings of the 4th International Conference on Intelligent Control and Information Processing(ICICIP), Beijing, China, 9–11 June 2013.
12. Liu, J.; Chung, S.H. An efficient load balancing scheme for multi-gateways in wireless mesh networks. *J. Inf. Process. Syst.* **2013**, *9*, 365–378.
13. Sinha, A.; Lobiyal, D.K. Performance evaluation of data aggregation for cluster-based wireless sensor network. *Hum.-Centric Comput. Inf. Sci.* **2013**, doi:10.1186/2192-1962-3-13.

14. Yoon, M.; Kim, Y.K.; Chang, J.W. An energy-efficient routing protocol using message successrate in wireless sensor networks. *J. Converg.* **2013**, *4*, 15–22.
15. Telit_Wireless M-Bus User Guide Part4 + Part5, Mode R2. Available online: http://teleorigin.com (accessed on 6 May 2013)