Noname manuscript No.
(will be inserted by the editor)

Lifetime Extension Based on Residual Energy for Receiver-driven Multi-hop Wireless Network

Chuluunsuren Damdinsuren · Daichi Kominami · Masashi Sugano · Masayuki Murata · Takaaki Hatauchi

Received: date / Accepted: date

Abstract An important research topic in wireless sensor networking is the extension of operating time by controlling the power consumption of individual nodes. In a receiver-driven communication protocol, a receiver node periodically transmits its ID to the sender node, and in response the sender node sends an acknowledgment, after which data transmission starts. By applying such a receiver-driven protocol to wireless sensor networks, the average power consumption of the network can be controlled, but there still remains the problem of unbalanced load distribution among nodes. Therefore, part of the network shuts down when the battery of the node that consumes the most power is completely discharged. To extend the network lifetime, we propose a method where information about the residual energy level is exchanged through ID packets in order to balance power consumption. Simulation results show that the network lifetime can be extended by about 70-100% while maintaining high network performance in terms of packet collection ratio and delay.

Keywords Energy-efficient MAC protocol · Multi-hop wireless networks · Load balancing · Network lifetime

Chuluunsuren, D., Kominami, D., Murata, M.
Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita-shi, Osaka 565-0871, Japan
E-mail: {c-damdin, d-kominami, murata}@ist.osaka-u.ac.jp

Sugano, M.
School of Comprehensive Rehabilitation, Osaka Prefecture University
3-7-30 Habikino, Habikino-shi, Osaka 583-8555, Japan
E-mail: sugano@rehab.osakafu-u.ac.jp

Hatauchi, T.
Fuji Electric Systems Co., Ltd.
Fujicho, Hino-shi, Tokyo 191-8502, Japan
E-mail: hatauchi-takaaki@fujielectric.co.jp

1 INTRODUCTION

Saving energy in the operation of sensor nodes with limited battery life is an important problem that still awaits resolution in the development of wireless sensor networks. Various approaches to saving energy have been developed, for example, miniaturization of sensor nodes, the application of MAC protocols with sleep control and multi-hop routing [15]. In this paper, we use the MAC layer approach based on control of the intermittent operation. Since sleeping nodes consume considerably less energy than idling nodes, sensor nodes enter the sleep state when they are not sending or receiving data in order to reduce energy consumption. However, in such cases, nodes must be able to control the time intervals at which they wake up in order to communicate with one another. Control methods for intermittent operation are classified into two types: synchronous [14, 4] and asynchronous [1, 11, 8, 5, 13, 6].

Synchronous methods use a beacon to maintain synchronization between successive intermittent operations. The advantage of synchronization is the relatively short delay between the moment when a sensor node wakes up and data transmission starts, and the disadvantage is that the regular ID broadcasting consumes power and causes interference. In comparison, asynchronous methods can be either sender-driven or receiver-driven, depending on which type of node initiates communication. Although nodes can start communicating at any given time in either case, packet collisions must be controlled. Furthermore, in asynchronous intermittent operation, the sender node waits in an idle state until the receiver node awakens. For these reasons, in terms of reducing energy consumption, the synchronous method is advantageous in systems with periodic data collection and a comparatively high rate of data generation. Meanwhile, the asynchronous type of communication is superior in systems where packets are generated at a lower rate and more irregularly. In this study, we target applications where 1) data is generated at a low rate, 2) a high packet collection ratio is required and 3) the operating lifetime should reach several years without battery replacement. Hence, the most suitable type of communication for intermittently operating sensor nodes in an ad hoc network is based on an asynchronous receiver-driven method [9].

We proposed the intermittent receiver-driven data transmission (IRDT) protocol in our previous work [6]. This protocol was designed for an actual product under development, specifically, a battery-powered metering system which is expected to support prolonged operation. In IRDT, which is a receiver-driven asynchronous protocol, receiver nodes send their own IDs to inform other nodes that they are ready to receive packets. A sender node waits for an appropriate receiver node to send its ID, after which it establishes a link with the receiver and transmits a data packet. There is no occupancy of the channel in IRDT. Moreover, a sender node can select a receiver from one or more communication candidates, thereby improving the reliability and reducing the time spent by an active sender in waiting for an appropriate receiver to wake up. Although the IRDT protocol can reduce the average power consumption, differences in power consumption still exist between individual nodes. Therefore, the batteries of the nodes with the highest power consumption will discharge fairly quickly while the remaining charge in other nodes will be disproportionately large. In that case, even though the total power remaining in the nodes is not zero, some parts of the network will shut down.

Therefore, a load balancing method that takes into account information about the residual energy is necessary in order to solve the problem described above. Considerable research has been conducted on load balancing based on information about the residual energy. For instance, in one approach, the entire network is divided into clusters, and routing to the sink is calculated on the basis of either a clustering method [16, 7, 10] or factors such as the cost of power consumption and information about the remaining charge. In the latter case, the cluster heads are set according to the residual energy. Another approach is to use a routing method [2, 12, 3, 17] to calculate the most appropriate route. However, there is limited research on techniques for controlling the intermittent interval on the basis of information about the residual energy in node batteries. In this paper, we propose one such method, which successfully extends the lifetime of the network. Our approach is intended for IRDT-based multi-hop wireless networks, and implements minimal modifications and additions to the existing IRDT method with almost no additional overhead. In this method, each node broadcasts information about its own residual energy to nearby nodes through ID packets, and all nodes control their own intermittent interval by using this information received from neighbor nodes, thus implementing efficient load balancing.

The rest of this paper is organized as follows. In Section 2, we discuss related work on duty-cycle MAC protocols for sensor networks. Section 3 presents an outline of the basic operation of IRDT and the methods employed in its implementation. Section 4 presents the details of the proposed method for load balancing, and Section 5 presents an evaluation of the proposed method conducted as a computer simulation. Finally, Section 6 presents the conclusion of this work together with a discussion about the direction of future research.

2 RELATED WORK

Existing contention-based duty-cycle MAC schemes in wireless sensor networks can be divided into two categories: synchronous and asynchronous.

2.1 Synchronous MAC Protocols

In synchronous MAC implementations, such as S-MAC [14] and T-MAC [4], scheduling information which specifies the cycles of the active and sleep periods is shared via synchronization packets. These packets align the active and sleep intervals of neighbor nodes, which wake up only during the common active time intervals to exchange packets. Since the active intervals are usually short, substantial amounts of energy can be saved. However, energy is still wasted since the strict synchronization of the clocks of neighbor nodes imposes high overhead.

In S-MAC [14], at the beginning of the active period, nodes exchange SYNC information with their neighbors to ensure that the node and its neighbors wake up concurrently. This schedule is only adhered to locally, resulting in a virtual cluster, which mitigates the need for synchronization of all nodes in the network. Nodes that lie on the border between two clusters maintain the schedules of both clusters, thus maintaining connectivity across the network. After the synchronization information is exchanged, the nodes send data packets by using request-to-send (RTS)/clear-to-send (CTS) signals until the end of the active period, after which they enter the sleep state. S-MAC saves energy with the use of periodic sleep intervals and reduces the amount of energy wasted as a result of idle listening. Although S-MAC improves the energy efficiency, it causes delays in multi-hop data delivery and wastes energy due to fixed duty cycling.

In T-MAC [4], which is an improvement on S-MAC, nodes enter sleep state when no activation event has occurred for a certain amount of time, which enhances the energy efficiency. In contrast to S-MAC, it operates with fixed-length slots and uses a time-out mechanism to dynamically determine the end of the active period. The time-out value (TA) is set to span a small contention period and the time necessary for an RTS/CTS exchange. If a node does not detect any activity within the time-out interval, it assumes that no neighbor is ready to communicate with it, and enters the sleep state. T-MAC adapts to traffic fluctuations in the network and improves the energy efficiency more drastically than S-MAC. However, a certain amount of energy is still wasted due to the fixed duty cycle in the fixed time-out mechanism.

2.2 Asynchronous MAC Protocols

In asynchronous MAC protocols, on the other hand, nodes do not exchange synchronization information for transmission or reception of data, thus enabling nodes to operate with their own independent duty cycles. Therefore, such protocols are not burdened by the overhead associated with the synchronization process. Asynchronous MAC protocols can be divided into two subcategories: sender-driven MAC protocols (such as B-MAC [11], WiseMAC [5], and X-MAC [1]) and receiver-driven MAC protocols (such as RI-MAC [13] and IRDT [6]). First, we introduce some classic sender-driven protocols.

B-MAC [11] utilizes a long preamble to achieve low power consumption during communication. If a node is ready to send data, first it sends a preamble which is slightly longer than the sleep period of the receiver. During the active period, the receiver node samples the channel, and if a preamble is detected, it remains awake to receive the data. With the inclusion of a long preamble, the sender ensures that at some point during the preamble the receiver will wake up, detect the preamble, and remain awake in order to receive the data. While B-MAC performs rather well, it suffers from an overhearing problem, in that receivers that are not the target of the sender also wake up during the long preamble and remain awake until the end of the preamble in order to find out if the packet is destined for them. This process wastes energy for all non-target receivers within the transmission range of the sender, and thus the long preamble dominates the energy consumption and increases the per-hop latency.

The method in WiseMAC [5] is similar to that in B-MAC, with the difference that the sender learns the wake-up periods of the receivers and schedules its transmissions in a manner that reduces the length of the extended preamble. The size of the preamble is initially set to be equal to the sampling period. However, the receiver might not be ready at the end of the preamble; as a consequence, factors such as interference may result in energy waste due to overemitting. Moreover, over emitting increases proportionally to the length of the preamble and the data packet since there is no handshake between the sender and the target receiver.

In contrast to B-MAC, X-MAC [1] utilizes short preambles with target address information instead of a long preamble, which solves the overhearing problem in B-MAC. When a receiver wakes up and detects a short preamble, it checks the target address included in the preamble. If the node is the intended receiver, it remains awake for the incoming data; otherwise, it immediately re-enters the sleep state. This mechanism can decrease the per-hop latency and the amount of energy wasted on waiting for data transmission. This approach is simple to implement and achieves low power consumption for communication, but becomes less energy efficient and fails to guarantee a worst-case delay as the traffic load increases.

A different approach is adopted in RI-MAC [13] and IRDT [6], which are receiver-driven MAC protocols. Contrary to sender-driven protocols, where the sender initiates the data transmission, RI-MAC offers receiver-initiated listening and low power consumption. The goal of RI-MAC is to reduce the channel occupancy time, which is achieved by ensuring that the sender remains active and waits silently until the receiver sends a short packet to explicitly signal that it is ready for data transmission. Since only such short packets and the transmitted data occupy the medium in RI-MAC, there are no preamble transmissions as in B-MAC and X-MAC, and the occupancy of the medium is drastically decreased, allowing other nodes to exchange data during that time.

3 OUTLINE OF IRDT

3.1 MAC Protocol

In the IRDT protocol, each node intermittently sends its own ID over the network, and then assumes a “receive wait” state for a short time before entering the sleep state. In the receive wait state, the sender node waits for the ID of an appropriate receiver node, and if such an ID arrives, the sender responds with a send request (SREQ) packet. After receiving an acknowledgement (RACK) packet for the SREQ packet, the sender transmits a data packet and ends communication following receipt of an acknowledgement packet for the data (DACK). Such intermittent operation involving Receiver 1 and Receiver 2 is shown in Fig. 1. Here, the sender node checks the ID of Receiver 2 and accepts it as an appropriate receiver. The appropriateness of a receiver is determined on the basis of the routing protocol (Section 3.2). The sender node can choose from one or more communication candidates, thus improving the communication reliability and reducing the time spent by the sender waiting for receivers to wake up.

3.2 Routing Protocol

The routing algorithm of IRDT is based on multi-hop routing, where each node is involved in the process of relaying

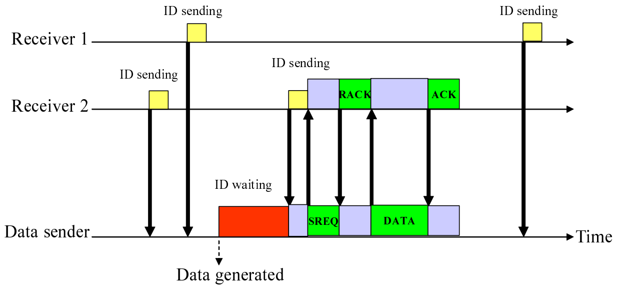


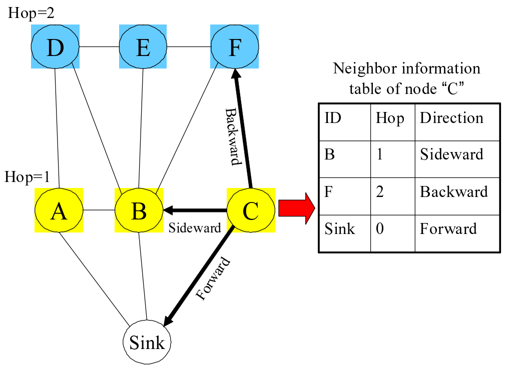
Fig. 1 Data transmission process in the IRDT protocol

the packet. The routing mechanism of IRDT is minimum-hop routing, because of simplicity and energy-efficiency [6]. Although minimum-hop routing can achieve the lowest energy consumption for delivering a data to sink, in some situations certain nodes cannot be used for minimum-hop routing, owing to poor conditions for radio wave transmission or node failure. Therefore, for higher flexibility, the routing algorithm of IRDT considers alternative paths in addition to the minimum-hop route. All nodes contain a configuration table for managing neighbor node information. Nodes update their own tables by periodically exchanging ID packets, and they use their own tables to select the packet forwarding node. Here, ID packets contain sender node's identification number and sender node's hop number to sink node.

If the minimum number of hops from a node to the sink node is denoted as h, the number of hops for any of its neighbor nodes is $h - 1$, h, or $h + 1$, and we refer to these neighbors as forward nodes, sideward nodes, and backward nodes, respectively. For example, regarding node "C" in Fig. 2, "Sink" is a forward node, "B" is a sideward node, and "F" is backward nodes. For minimum-hop routing, the sender node should select forward nodes as receivers. When a sender receives the ID of a forward node, it returns an SREQ packet. We define communication failure as a situation where the sender cannot obtain RACK and DACK packets from the receiver. Sideward nodes are selected when communication failure has occurred with all forward nodes, and backward nodes are selected if communication fails with all sideward nodes. All data packets contain a time-to-live (TTL) field in order to avoid heavy repetition of data relay. For each relay of a data packet, TTL is decremented by one, and when TTL becomes 0, the data packet is discarded. A node will not select a sideward node or a backward node if doing so would result in data packet loss due to the TTL mechanism.

3.3 Problems with IRDT

IRDT has been demonstrated to be capable of controlling the average power consumption in a network where data is generated at a low rate. However, differences in power consumption exist between nodes as a result of differences in loading due to topology. Therefore, part of the network shuts down when the remaining charge of the nodes with the highest power consumption becomes zero. The first time when the battery of a node is completely discharged is defined as the network lifetime in this paper. Therefore, the power consumption of the nodes with the highest power consumption must be controlled in order to maximize the network lifetime, preferably by balancing the load between nodes and leveling the power consumption. In this paper, a method for extending the network lifetime by controlling the intermittent interval according to the residual energy is proposed.



ID	Hop	Direction
B	1	Sideward
F	2	Backward
Sink	0	Forward

Fig. 2 Management of neighbor information table

4 LOAD BALANCING BY CONTROLLING INTERMITTENT INTERVAL

4.1 Basic Methodology

In the IRDT protocol, every node periodically broadcast a packet that include its ID. Therefore, if this mechanism is used efficiently, information can be transmitted between nodes in a manner that maximizes the network lifetime. For this purpose, a node includes residual energy information in an ID packet and broadcasts it to neighbor nodes. When a node receives the ID packet, it updates its local information about the residual energy of the sender node in the neighbor information table. A mechanism for handling such information can be achieved by implementing a small change in the existing IRDT method, with negligible additional overhead (Fig. 3).

Next, we describe a novel approach for load balancing which is based on the control of the intermittent interval. The relay load of individual nodes can be adjusted by changing their respective intermittent intervals since a longer intermittent interval has a smaller probability of relaying packets from other nodes.

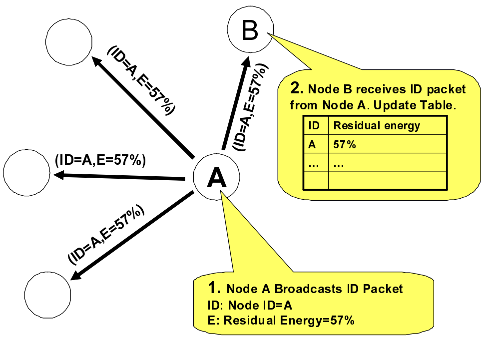


Fig. 3 Node A periodically broadcasts ID packets containing information about its own residual energy. Upon receiving an ID packet from node A, node B updates its information table with the new value of the residual energy.

Table 1 Description of parameters and variables in pseudocode for node i

Parameter&Variable	Description
$E[i]$	Residual energy
$Esw[i]$	Avg residual energy of all sideward nodes
$T[i]$	Intermittent interval
$N_{sideward}$	Set of sideward nodes
α	Non negative constant value
δ	Random value
T_{max}	Upper bound of the intermittent interval
T_{min}	Lower bound of the intermittent interval
$table_i$	Neighbor information table

In the proposed load balancing method, the intermittent intervals of individual nodes are controlled by using their residual energy relation to their sideward nodes. In other words, each node controls its own intermittent interval according to the difference between its own residual energy and the average residual energy of all of its sideward nodes. Here, a node and all of its sideward nodes have the same hop count. Since the packet relaying load depends on the node's hop count, a different hop count results a large difference in power consumption for packet relaying. Therefore, our method only consider load balancing between nodes which have same hop count. We show that it is possible to implement load balancing for nodes with the same hop count.

4.2 Details of Algorithm

In particular, each node updates its own intermittent interval according to the algorithm described below. The algorithm consists of three parts: 1) broadcasting information about the residual energy, 2) maintaining a neighbor information table, and 3) controlling the intermittent interval. All parameters and variables are described in Table 10.

1. **Broadcasting Information about the Residual Energy**
   At the beginning of each intermittent interval, node i broadcasts an ID packet that includes its ID and information about its residual energy.

```
$ID_{send}$=GenerateID($i$,$E_i$);
Send($ID_{send}$);
```

2. **Maintaining Neighbor Information Table**
   When node i receives an ID packet from a neighbor node, it reads the information about the residual energy of the sender node and records the new information into the neighbor information table.

```
$table_i[ID_{receive}.id] = ID_{receive}.battery$
```

3. **Controlling Intermittent Interval**
   Node i renews its intermittent interval every 100 s, which is an experimentally obtained value. In our simulation, the average packet transmission rate is 1 packet per 100 s. Therefore, we assume that the average interval between successive updates of the residual energy is also 100 s. The average residual energy of all sideward nodes is calculated as blow.

```
$E_{sw}[i] = \frac{1}{|N_{sideward}|} \cdot \sum_{s \in N_{sideward}} table_i[s]$
```

   Next, node i compares its residual energy with the average residual energy of all sideward nodes. If the charge of node i is larger, then it shortens its own intermittent interval, and otherwise, lengthens it.

```
if ($E[i] \geq E_{sw}[i]$)
        $T[i] = T[i] - \alpha + \delta$
else
        $T[i] = T[i] + \alpha + \delta$
```

   Finally, node i controls its intermittent interval according to the constraints listed below. Thus, on each update, the node ensures that the intermittent interval satisfies the conditions for the upper and lower bounds.

```
$T_{min} \leq T[i] \leq Tmax$
```

Furthermore, if the intermittent intervals of neighbor nodes synchronize, the times at which nodes transmit packets overlap, resulting in repeated collisions. For this reason, we add a random component $\delta(= [-0.04, +0.04]$ s) to the intermittent intervals in order to prevent such synchronization.

The operation stability is very important under various situations such that very big battery deviation, or intermixed conditon of new nodes and old nodes. Furthermore, since the sensor nodes do not have enought processing capability, we would like to control by the algorithm as simple as possible and same at every sensor nodes. Then, it is possible that it can also save power required for control. Therefore, our proposed method changes the intermittent interval by tha constant value α. The constant α defines the amount by which an intermittent interval is increased or decreased when there is a difference in the residual energy. When α is too large, the intermittent interval is changed abruptly, and when α is too small, the effect of the control mechanism becomes small. Therefore, α should be set such that the amount of increase or decrease is optimal.

T_{min} denotes the lower bound of the intermittent interval; a node should not operate at intervals shorter than this lower bound. For nodes with low power consumption, the proposed mechanism increases the intermittent interval of the node, providing sufficient time in the sleep state. However, when the intermittent interval of a receiver node is large, sender nodes might face the problem of being unable to send their data. This eventually reduces the rate of data collection at the sink while also increasing the power consumption at sender nodes, as they must wait for ID packets from receiver nodes. Therefore, we set T_{max} as an upper bound of the intermittent interval in order to prevent the occurrence of the abovementioned problem.

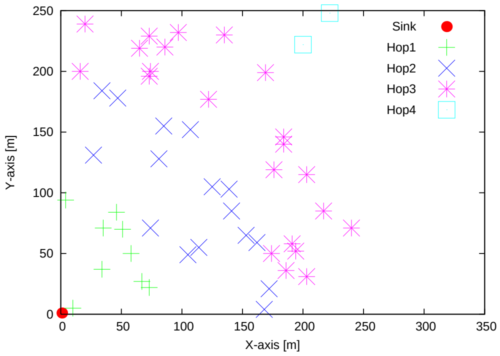
By applying the proposed technique, when the residual energy of a node is larger than the average residual energy of all of its sideward nodes, that node begins to relay more packets by shortening its own intermittent interval. Conversely, when the residual energy of a node is small, it spends more time in the sleep state by lengthening the intermittent interval. Each sensor node increases and decreases its own intermittent interval according to this principle, thus preventing bias in the power consumption distribution.

5 SIMULATION RESULTS

In this section, we evaluate the performance of the proposed method through a computer simulation. We assume networks with the topology shown in Fig. 4. “Sensor” node number is 49, and “Sink” node number is 1. Sensor nodes transmit packets to the sink node randomly in accordance with a predefined rate. Our target application is auto gas metering system, therefore we used this grid topology. Furthermore, randomly deployed topology is used. The main parameters of this setup are shown in Table 3. In Fig. 4, the nodes with the same number of hops are shown in the same color. Detailed topology information is shown in Table 2.

Table 2 Topology Information

Parameter	Value
Transmission range	10 m
Vertical distance of sensors	2.5 m
Horizontal dostance of sensors	2.5 m



(a) Topology A: A random deployed topology. The left bottom is the sink node.

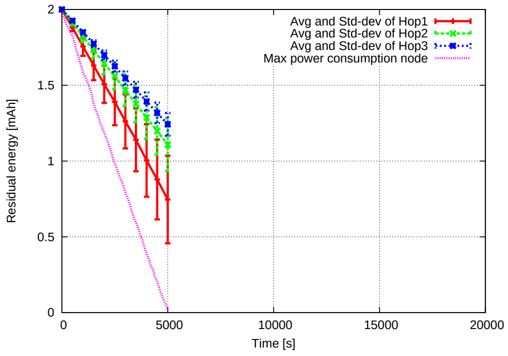
(b) Topology B: The center is the sink node.

(c) Topology C: The left bottom is the sink node.

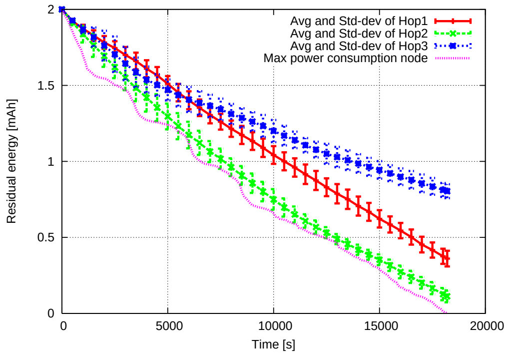
Fig. 4 Simulation topology

Table 3 Simulation Parameters

Parameter	Value
Packet generation rate	0.01 packets/s
Initial battery capacity	2 mAh
Initial intermittent interval	0.15 s
Current during transmission	20 mA
Current in the waiting state	25 mA
Current during reception	25 mA
Current in the sleep state	0 mA
Transmission rate	100 kbps
Packet size	128 bytes

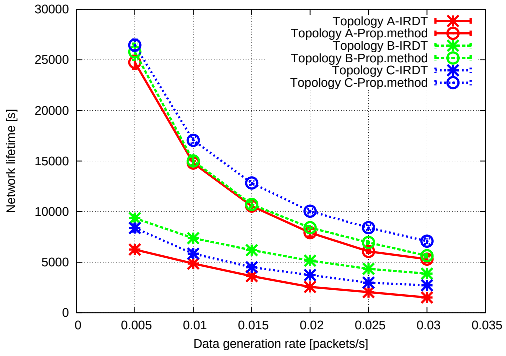


(a) Original IRDT: Changes in the residual energy. (Topology C)

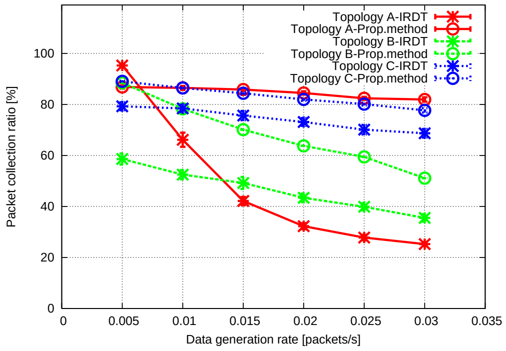


(b) Proposed improvement: Changes in the residual energy. The parameter values for ”Proposed method” are $T_{min} = 0.15$ s, $T_{max} = 1.5$ s, $\alpha = 0.15$ s. (Topology C)

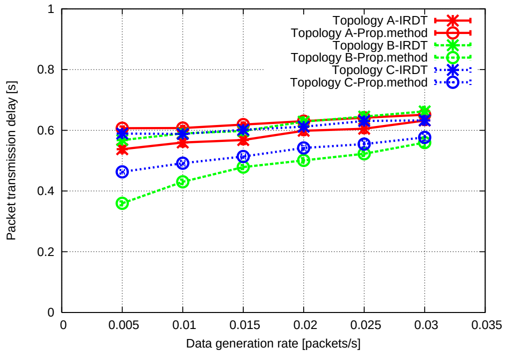
Fig. 5 Comparison of the change in the residual energy of nodes.



(a) Network lifetime



(b) Packet collection ratio



(c) Packet transmission delay

Fig. 6 Comparison of basic network performance parameters between original and proposed methods. The parameter values for ”Proposed method” are $T_{min} = 0.15$ s, $T_{max} = 1.5$ s, $\alpha = 0.15$ s.

5.1 Simulation Results for Basic Network Performance

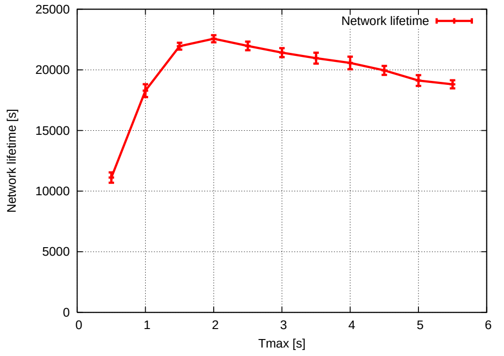
Figure 5 shows the changes in the residual energy for nodes in each hop number. This figure shows the average and the standart deviation of residual energy of nodes. The residual energy is leveled by implementing appropriate control taking into account the number of hops to the sink and the rate of change of the residual energy of the nodes with the highest power consumption. For the original IRDT, the residual energy linearly decreases. Since there are considerable differences in the loading of different nodes, and since the intermittent interval is constant at 0.15 s, when the network lifetime is reached due to the complete discharge of the battery of node 33, the batteries of many other nodes remain fairly full. When the proposed technique is applied, the difference in power consumption between nodes with the same number of hops is controlled, and the corresponding residual energy levels become equal for each number of hops, which demonstrates the effectiveness of the proposed method in extending the network lifetime. A comparison of the basic network performance for different packet generation rates is shown in Fig. 6. We compared IRDT with proposed method for each topology in Figure 4. The proposed method hardly affects the packet collection ratio and delay time, even though the network life time has been extended by about 75-100%. This is because the utilization rate is maintained for all nodes with a certain number of hops by controlling their respective intermittent intervals according to the residual energy in each node. In random topology, when packet generation rate increases, packet collection ratio and delay time is falling quickly, because of unbalanced load. However, the results show that our proposed method improve network lifetime, delay, and packet collection ratio. The difference between the center sink and the left bottom sink is that packet collision frequency at the sink node. If the sink node is deployed in the center, the 1 hope node number increase. Therefore, collision probability around sink is increase greatly. Packet collection ratio differences in the center sink and the left bottom sink are about 20%.

5.2 Analysis of Control Parameters

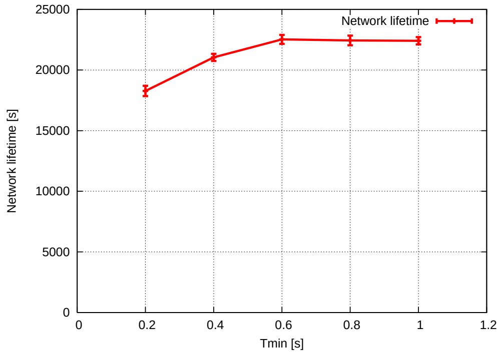
We show the results for network lifetime, packet collection ratio, and packet delay with respect to each parameter in proposed method in Figs. 7, 8, and 9, respectively. It can be seen in Fig. 7(a) that the upper bound T_{max} of the intermittent interval exerts strong influence on the network lifetime, as well as that an optimal value exists (about 2.0 s in this figure). Moreover, T_{max} is effecting greatly on packet collection ratio and delay time. But, T_{min} and α have only little affects on this metrics. The network lifetime for various values of the lower bound T_{min} of the intermittent interval and the constant α are shown in Fig. 7(b) and 7(c). These results indicate that the influence of these control parameters on the network lifetime is negligible. Figure 7(b) shows the tendency of the network lifetime to increase, and therefore the value of T_{min} is large. However, if an even larger value is set for T_{min}, the network cannot process the load, the packet collection ratio decreases, and the packet delay increases.

These results indicate that the network lifetime changes considerably depending on the upper bound T_{max} of the intermittent interval. After obtaining these results, we attempted to derive the optimal values for the control parameters by analyzing the theory behind the proposed method, as it was clarified that the upper bound T_{max} of the intermittent interval exerts strong influence on the network lifetime. In a sensor network, the highest load is concentrated at the sink node, and that power consumption is largest at the one-hop and two-hop nodes. The value of T_{max} that can minimize the power consumption of the node with the highest discharge rate is the most appropriate value in terms of prolonging the network lifetime. Therefore, if $E_{total}[i]$ denotes the total power consumption of node i, the optimal value of T_{max} can be expressed as follows.

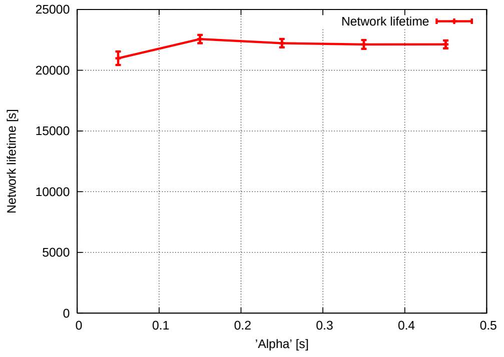
$$T_{max} = \arg\min \left\{ \max_{i \in \mathrm{Node}} \{E_{total}[i]\} \right\} \quad (1)$$



(a) T_{max}: Upper bound of the intermittent interval ($T_{min} = 0.6$ s, $\alpha = 0.15$ s)

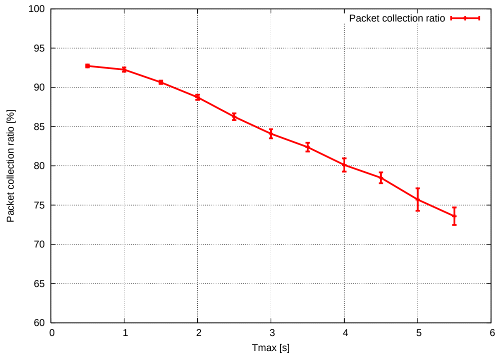


(b) T_{min}: Lower bound of the intermittent interval ($T_{max} = 2.0$ s, $\alpha = 0.15$ s)

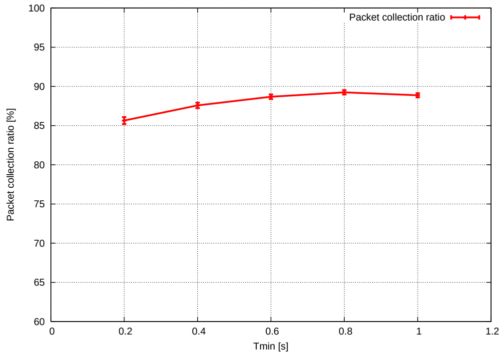


(c) α: Amount of increase or decrease in the intermittent interval ($T_{min} = 0.6$ s, $T_{max} = 2.0$ s)

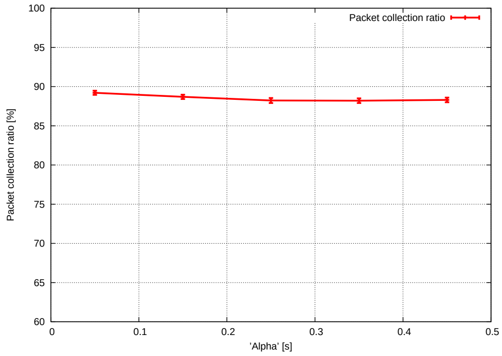
Fig. 7 Influence of three control parameters on the network lifetime. (Topology C)



(a) T_{max}: Upper bound of the intermittent interval ($T_{min} = 0.6$ s, $\alpha = 0.15$ s)

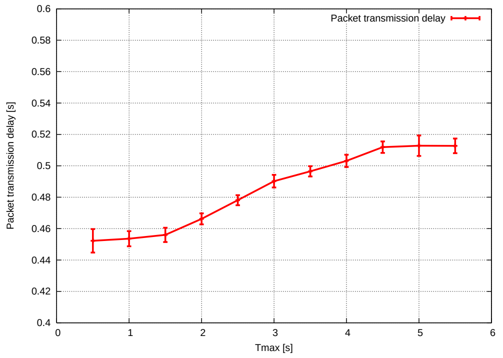


(b) T_{min}: Lower bound of the intermittent interval ($T_{max} = 2.0$ s, $\alpha = 0.15$ s)

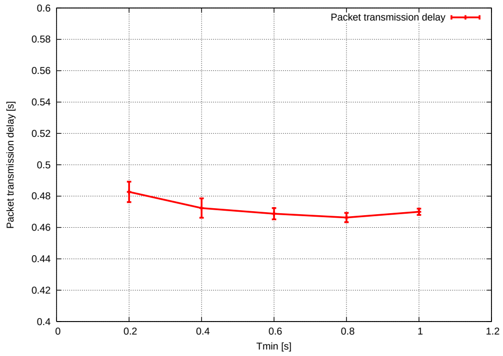


(c) α: Amount of increase or decrease in the intermittent interval ($T_{min} = 0.6$ s, $T_{max} = 2.0$ s)

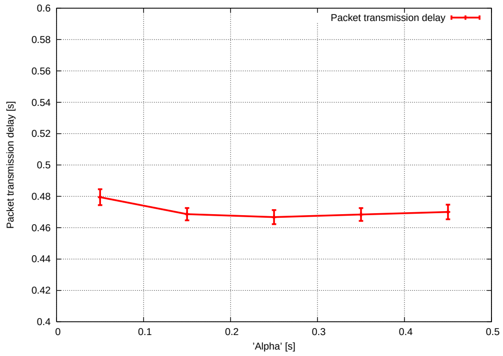
Fig. 8 Influence of three control parameters on the packet collection ratio. (Topology C)



(a) T_{max}: Upper bound of the intermittent interval ($T_{min} = 0.6$ s, $\alpha = 0.15$ s)



(b) T_{min}: Lower bound of the intermittent interval ($T_{max} = 2.0$ s, $\alpha = 0.15$ s)



(c) α: Amount of increase or decrease in the intermittent interval ($T_{min} = 0.6$ s, $T_{max} = 2.0$ s)

Fig. 9 Influence of three control parameters on the packet transmission delay. (Topology C)

First, we express the total amount of energy consumed by a node per unit time, after which we minimize this value in order to optimize T_{max}. The parameters used in this procedure are as follows: N_a is the number of ID packets arriving from the forward nodes per unit time, N_g is the number of packets generated per unit time, E_{rec} is the energy consumed during the reception of a packet, E_{send} is the energy consumed during the transmission of a packet, and E_{ID} is the energy consumed during ID transmission which is not used for data transmission.

Power consumption per unit time for data reception: If e_{rec} denotes the energy consumed during the reception of a single data packet, the total power consumption for packet reception is as follows.

$$E_{rec} = e_{rec} \cdot N_a \tag{2}$$

Power consumption per unit time for data transmission: e_{wait} is the energy consumed by a node in the ID waiting state, and e_{send} is the energy consumed during the transmission of a single data packet. Here, the average ID waiting time is T_{wait}. The energy consumed per unit time for data transmission is given as follows.

$$E_{send} = (T_{wait} \cdot e_{wait} + e_{send}) \cdot (N_a + N_g) \tag{3}$$

Power consumption for ID transmission: The energy consumed per unit time during ID transmission without data reception is given by the following expression, where t_{avg} is the average intermittent interval of the nodes. e_{id} denotes the energy consumed during the sending of a single ID packet. t_{id} is assumed to be the maximum of the ID waiting time.

$$E_{ID} = \left(\frac{1}{t_{avg}} - N_a\right) \cdot (e_{id} + e_{wait} \cdot t_{id}) \tag{4}$$

Total energy consumption: When the abovementioned expressions are integrated, the total energy consumed per unit time for a given node is as follows.

$$E_{total} = E_{rec} + E_{send} + E_{ID} \tag{5}$$

Analysis of average ID waiting time T_{wait}: Here, node A is in ID waiting state with respect to forward nodes F_i (Fig. 10). t_i is the average ID transmission interval of nodes F_i. Here, we focus on the average time passing until node A obtains an ID packet from one of its forward nodes for the first time.

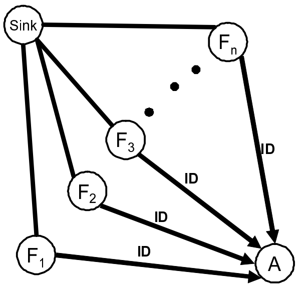


Fig. 10 F_i are all forward nodes of Node A. Node A is waiting ID from a forward node.

The time until node F_i transmits its ID is denoted by the probability variable x_i, which takes a uniform random value in the interval $[0 : t_i]$.

In order to simplify this analysis, we introduce such an assumption of random uniform distribution. Under the influence from adjacent nodes, actual ID transmission interval of each node is not uniform distribution. However, we show that the optimal value of control parameter T_{max} can be obtained approximately only by grasping a rough tendency by introducing such assumption. We show the results in the next paragraph. Furthermore, the ID waiting time is expressed as $z = min\{x_1, x_2, .., x_n\}$, and the average ID waiting time can be expressed as the expected value of a random variable z, which can be calculated as follows.

Distribution function:

$$\begin{aligned} F(u) = P(z \le u) &= 1 - P(z > u) \\ &= 1 - \prod_{1}^{n} P(x_i > u) \\ &= 1 - \prod_{1}^{n} \left(1 - \frac{u}{t_i}\right) \end{aligned} \tag{6}$$

Density function:

$$f(u) = \frac{dF(u)}{du} \tag{7}$$

Expected value:

$$E = \int_0^{+\infty} u \cdot f(u)\, du = \int_0^{+\infty} u \cdot \frac{dF(u)}{du} du \tag{8}$$

This procedure can be simplified as follows, where $t_m = min\{t_1, t_2, .., t_n\}$.

$$T_{wait} = t_m \cdot F(t_m) - \int_0^{t_m} F(u)\, du \tag{9}$$

Comparison of analysis and simulation results: The highest load in the network is concentrated at nodes 16 and 33, as shown above. Then, it is necessary to simply calculate the power consumption from Eq. (5) for these nodes

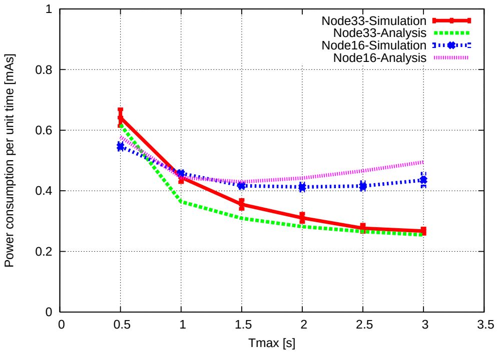
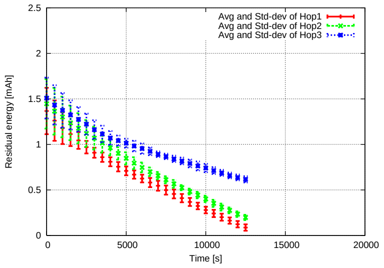


Fig. 11 Comparison of analysis and simulation results for T_{max}

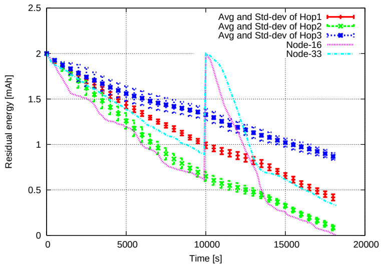
and to set T_{max} in such a way that the power consumption becomes minimal. Figure 11 shows the comparison between the above analysis results and the simulation results for the power consumption of nodes 16 and 33. As shown in this figure, the analysis results are mostly in agreement with the changes in the power consumption of nodes 16 and 33, and the optimal value of T_{max} as obtained from Eq. (5) is close to T_{max}=2.0 s as obtained from the simulation. However, there is an error margin between the simulation and the analysis since the influence of surrounding nodes has not been considered in this analysis.

5.3 Robustness of Proposed Method

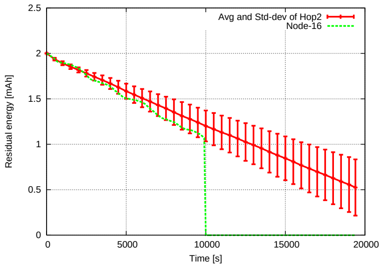
The term “robustness” refers to the ability of the network to continue operating when initial battery capacities of nodes are uneven or when the battery of a node is replaced. Moreover, we shows the simulation results of node’s failure and a new node’s addition. The simulation results are shown in Figure. 12. Figure 12(a) shows the results of a simulation where the initial charge of all batteries was set random value between 1-2 mAh, and it can be seen from these results that the residual energy of the nodes converge with time. Figure 12(b) shows the simulation results for the case where the batteries of nodes 16 and 33 were replaced with fresh batteries at 10000 s. It can be confirmed that while transmission from other nodes of hops 1 and 2 though nodes 16 and 33 gradually becomes slower, it is accelerated immediately after the batteries are replaced. When the residual energy of nodes 16 and 33 becomes equivalent to that of other nodes, they continue operating with almost the same intermittent cycles. Thus, the network lifetime can be extended by adding a new node in a part of the network where the average residual energy of the nodes has decreased drastically. Figure 12(c) and Figure 12(d) shows simulation results for the case where node’s failure in the network, and the new node’s addition into the network. We confirmed that network operates without any problems.



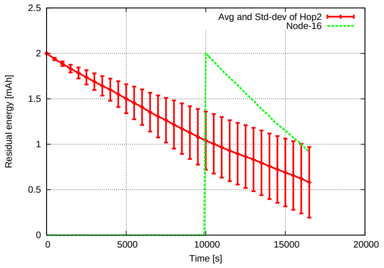
(a) Results in the case of large variation in the initial charge of the batteries



(b) Replacement of the batteries of nodes 16 and 33 at 10000 s



(c) The node 16 failured at 10000 s



(d) The node 16 added at 10000 s

Fig. 12 Robustness of the proposed method. The parameter values for “Proposed method” are $T_{min} = 0.15$ s, $T_{max} = 1.5$ s, $\alpha = 0.15$ s. (Topology C)

6 CONCLUSION

In this paper, we proposed a technique for extending the lifetime of wireless receiver-driven multi-hop networks. The method implements load balancing by allowing each node to control its own intermittent interval based on the difference between its own residual energy and the average residual energy of all of its sideward nodes. Through simulation, it was demonstrated that this method can be used for extending the network lifetime by about 70-100% while maintaining high network performance. Moreover, further analysis was conducted in order to derive the optimal value for the upper bound T_{max} of the intermittent interval, which exerts strong influence on the network lifetime. Future work will be oriented towards conducting more accurate analysis by considering the influence of surrounding nodes, and to confirm the applicability of the proposed method to various types of networks. Furthermore, we are considering the extension of this method to adaptive systems, which can be applied in cases where network load changes dynamically. And, we will do experiment that based on real testbed.

References

1. Buettner, M., Yee, G.V., Anderson, E., Han, R.: X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks. In: Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys 2006), pp. 307–320 (2006)
2. Chang, J.H., Tassiulas, L.: Maximum lifetime routing in wireless sensor networks. IEEE/ACM Transactions on Networking **12**(4), 609–619 (2004)
3. Chen, Y., Zhao, Q.: Maximizing the lifetime of sensor network using local information on channel state and residual energy. In: Proceedings of the 2005 Conference on Information Science and Systems (2005)
4. van Dam, T., Langendoen, K.: An adaptive energy-efficient MAC protocol for wireless sensor networks. In: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys 2003), pp. 171–180 (2003)
5. El-Hoiydi, A., Decotignie, J.: Wisemac: an ultra low power mac protocol for the downlink of infrastructure wireless sensor networks. In: Ninth International Symposium on Computers and Communications 2004 (ISCC 2004), vol. 1, pp. 244–251 (2004)
6. Hatauchi, T., Fukuyama, Y., Ishii, M., Shikura, T.: A power efficient access method by polling for wireless mesh networks. IEEJ Transactions on Electronics, Information and Systems **128**(12), 1761–1766 (2008)
7. Israr, N., Awan, I.: Coverage based inter cluster communication for load balancing in heterogeneous wireless sensor networks. Telecommunication Systems **38**(3), 121–132 (2008)
8. Jurdak, R., Baldi, P., Lopes, C.V.: Adaptive low power listening for wireless sensor networks. IEEE Transactions on Mobile Computing **6**(8), 988–1004 (2007)
9. Kominami, D., Sugano, M., Murata, M., Hatauchi, T., Fukuyama, Y.: Performance evaluation of intermittent receiver-driven data transmission on wireless sensor networks. In: Proceedings of the 6th International Symposium on Wireless Communication Systems (ISWCS 2009), pp. 141–145 (2009)
10. Lee, S., Lee, J., Sin, H., Yoo, S., Lee, S., Lee, J., Lee, Y., Kim, S.: An energy-efficient distributed unequal clustering protocol for wireless sensor networks. World Academy of Science, Engineering and Technology **48**, 443–447 (2008)
11. Polastre, J., Hill, J., Culler, D.: Versatile low power media access for wireless sensor networks. In: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys 2004), pp. 95–107 (2004)
12. Shah, R.C., Rabaey, J.M.: Energy aware routing for low energy ad hoc sensor networks. In: Proceedings of IEEE Wireless Communications and Networking Conference, vol. 1, pp. 350–355 (2002)
13. Sun, Y., Gurewitz, O., Johnson, D.: RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. In: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys 2008), pp. 1–14 (2008)
14. Ye, W., Heidemann, J., Estrin, D.: An energy-efficient MAC protocol for wireless sensor networks. In: Proceedings of IEEE INFOCOM 2002, vol. 3, pp. 1567–1576 (2002)
15. Yick, J., Mukherjee, B., Ghosal, D.: Wireless sensor network survey. Computer Networks **52**, 2292–2330 (2008)
16. Younis, O., Fahmy, S.: HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. IEEE Transactions on Mobile Computing **3**(4), 366–379 (2004)
17. Zhang, H., Shen, H., Tan, Y.: Optimal energy balanced data gathering in wireless sensor networks. In: Proceedings of IEEE International Parallel & Distributed Processing Symposium 2007 (IPDPS 2007) (2007)