

## Design of a Reliable Real-Time Scheduling Policy for Dual-Channel Networks<sup>\*</sup>

JUNGHOOON LEE, MIKYUNG KANG, GYUNG-LEEN PARK AND SUNG Y. SHIN<sup>†</sup>

*Department of Computer Science and Statistics*

*Cheju National University*

*Jeju Do, 690-756 Korea*

*E-mail: {jhlee; mkkang; glpark}@cheju.ac.kr*

<sup>†</sup>*School of Computer Science and Engineering*

*South Dakota State University*

*South Dakota, SD 57007, U.S.A.*

*E-mail: sung.shin@sdstat.edu*

This paper designs and analyzes the performance of a communication scheduler that cooperates with each other over the dual-channel wireless network, aiming at enhancing reliability and timeliness of time-sensitive applications on embedded devices. The proposed scheme consists of (1) evenly partitioning the stream set into two identical ones, (2) applying EDF (Earliest Deadline First) policy on the respective set after an acceptance test, and (3) maximizing the number of switchable pairs. Based on this schedule and channel estimation, the coordinator schedules each node according to 3-level selection steps made up of normal schedule, channel switch, and slot reallocation. Simulation results show that the proposed scheme improves the deadline meet ratio by up to 28 %, compared with the global EDF policy, when packet error rate reaches 0.4. The slot reallocation can enhance the real-time performance by 11% when the deadline of each application gets larger up to 5 times of execution time.

**Keywords:** 3-level scheduling, embedded operating system, time-sensitive application, dual wireless channel, channel error handling

### 1. INTRODUCTION

Nowadays, for the embedded system connected via wireless network like IEEE 802.11 WLAN, its applications are typically time-sensitive and their time constraints can not be satisfied without an efficient communication mechanism. Operating systems such as Windows Mobile for low-resource real-time embedded system face severe constraints including the need to meet strict deadlines and also cope with low resource availability [1]. However, the wireless communication is subject to different type of errors, and such faults and errors may cause scheduling malfunctions and incorrect system behavior [2]. Even though the system is robust enough to continue its operation in the presence of some fault, error handling procedure imposes a significant overhead to the system performance. Hence, it is desirable to mask errors in each resource manager such as communication.

---

Received November 15, 2006; accepted February 15, 2007.

Communicated by Sung Shin and Tei-Wei Kuo.

<sup>\*</sup>This research was supported by the MIC, Korea, under the ITRC support program supervised by the IITA (IITA-2006-C1090-0603-0040).

Meanwhile, one of the promising application areas of wireless technology is the wireless sensor network [3], which is constructed by a group of sensors to perform tasks such as monitoring harsh environments, providing real-time information on fire detection, and so on [4]. In this environment, periodically sampled data are delivered to the remote controller application within a specified deadline so that it can determine a timely reaction. In other words, the message of sampled data has a real-time constraint that it should be transmitted within a bounded delay [5]. Otherwise, the data are considered to be lost, and the loss of a real-time message may jeopardize the correctness of execution result or system itself. Accordingly, a real-time message stream strongly demands the guarantee from the underlying network that its time constraints are always met prior to the system operation or connection setup via the reservation of a fraction of network bandwidth.

However, the guarantee scheme is not solely sufficient to satisfy such time constraints due to unpredictable location-dependent and bursty errors that make a real-time application fail to send or receive some of its packets [6]. In the mean time, the wireless network has an advantage that a cell is able to operate dual channels, as a cell can have up to 3 channels according to the IEEE 802.11 standard [7]. In this system, each mobile station can be equipped with two or more (transmitter, receiver) pairs, giving flexibility to transmit and receive simultaneously on both channels. After all, the dual channel system enables various ways to improve the reliability of the message delivery.

The dual network architecture is analogous to the dual processor system in that both network and processor can be considered as an active resource. However, real-time scheduling for dual or multiple resource systems is known to be an NP-hard problem [8], while the uniprocessor system has an optimal scheduling solution such as RM (Rate Monotonic) for static scheduling as well as EDF (Earliest Deadline First) for dynamic scheduling. Applying the RM or EDF method to multiple resource systems is not optimal in scheduling preemptable jobs due to its work conserving nature [9]. Existing multichannel scheduling schemes such as MULTI-FIT cannot be applied directly to real-time communication, as they don't directly take into account the timeliness requirement [8].

Even though the EDF scheme is not optimal in dual networks, it is sufficiently efficient in terms of schedulability as well as low runtime overhead. Particularly, in wireless sensor networks where each message usually has no data dependency, the advantage of the EDF scheme can be sustained also for the dual networks by evenly partitioning each stream. Moreover, the dual channels can cope with network errors without violating the time constraints of messages by dynamically switching the polling schedule across the channels according to current channel link status. With such assertions, this paper designs and analyzes the performance of a communication scheduler for the dual-channel wireless network, aiming at inheriting the efficiency of the EDF scheduling scheme as well as improving the capability of coping with wireless channel errors.

The rest of this paper is organized as follows: After introducing backgrounds and related works in section 2, section 3 specifies the scope of this paper and basic assumptions. Section 4 describes the proposed scheduler in detail along with a relevant example. After exhibiting the performance measurement result in section 5, section 6 finally concludes this paper and describes future works.

## 2. BACKGROUND AND RELATED WORKS

The IEEE 802.11 was developed as a MAC (Medium Access Control) standard for WLAN, and the standard consists of both an essential DCF (Distributed Coordination Function) and an optional PCF (Point Coordination Function) [10]. PCF and DCF are mapped into CFP (Contention Free Period) and CP (Contention Period), respectively. The DCF exploits collision-based CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) protocol for non-real-time messages, aiming at enhancing not only average delivery time but also overall network throughput. However, packet collisions, intrinsic to CSMA and its variants, make it impossible for a node to predictably access the network. In contrast, PCF mode makes one AP (Access Point) poll each station to determine which station has the right to transmit during a given time slice, providing the bounded latency for data transmission.

On the other hand, the available wireless spectrum is divided into several channels, so ability to utilize multiple channels substantially increases the effective bandwidth available to wireless network nodes [11]. The IEEE 802.11b standard specifies 11 channels operating in the 2.4 GHz band with 80 MHz of reusable spectrum [12]. Even though the channel overlap problem limits the maximum number of simultaneous channels to 3, it is possible to create multiple channels from the wireless spectrum in a cell.

As the most prominent dynamic priority scheduling mechanism for the uniresource real-time system, the EDF algorithm assigns priorities to individual jobs in tasks according to their absolute deadlines. In the area of wireless networks, M. Caccamo *et al.* have proposed a MAC that supports deterministic real-time scheduling via implementation of TDMA (Time Division Multiple Access) to apply EDF policy to the WLAN [13]. Referred as *implicit contention*, their scheme makes every station concurrently run the common real-time scheduling algorithm to determine which message can access the medium. As such, each message implicitly contends for the medium through the scheduling algorithm, for example, with priorities, instead of contending explicitly on the physical medium, thus it can save power and time.

Traditionally, there have been two approaches for scheduling periodic tasks in dual or multiprocessors, namely, *partitioning* and *global scheduling* [8]. The partitioning scheme assigns each stream to a single network, on which messages are scheduled independently. The main advantage of partitioning approaches is that they reduce a multiprocessor scheduling problem to a set of uniprocessor scheduling problems. However, finding an optimal assignment to networks is a bin-packing problem, which is NP-hard in the strong sense. In global scheduling, all eligible tasks are stored in a single priority-ordered queue while the global scheduler selects the highest priority task for execution from this queue according to the policy, such as EDF [14]. CASP (Contiguous Algorithm for Single Priority) maintains an allocation vector  $V$ , where  $V_i$  represents the partial sum of slots currently allocated to channel  $i$  [12]. For a given request, the scheduling algorithm allocates the request contiguously on the channel with the least partial sum of allocation. In contrast, NCASP (Non-continuous Algorithm for Single Priority) defines an overflow amount,  $\phi$ , and if an assignment makes  $V_i$  exceed  $\phi$ , it is split and the overflowed part is assigned to another resource. However, the method of deciding  $\phi$  brings about another complex case-sensitive problem.

### 3. BASIC ASSUMPTIONS

#### 3.1 Network and Error Models

Each cell is assumed to consist of an AP (or coordinator) and multiple MSs (Mobile Stations) as shown in Fig. 1. Each member of a cell has interfaces to two respective channels, and can transmit or receive data on two channels at the same time. AP may be linked to a wired backbone or other wireless channel to exchange intercell messages. In a cell, every station shares medium on the common frequency band and accesses according to the predefined MAC protocol. Whether each flow is an uplink (MS to AP) or downlink (AP to MS), it is transmitted under the control of the AP.

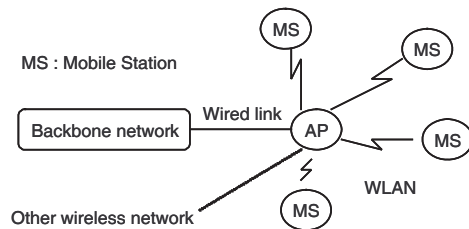


Fig. 1. Network model.

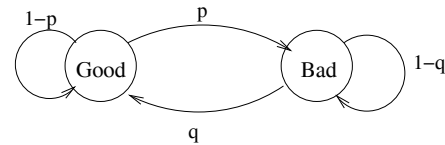


Fig. 2. Error model.

In a channel, each station is associated with a channel link which has either of two states, namely, *error state* and *error-free state* at any time instant. A channel link is defined between each mobile and the AP, and it can be modeled as a Gilbert channel [6]. According to this model, we can denote the transition probability from state *good* to state *bad* by  $p$  and the probability from state *bad* to state *good* by  $q$ , as shown in Fig. 2. The pair of  $p$  and  $q$  representing a range of link condition, has been obtained using trace-based channel link estimation. The average error probability and the average length of a burst of errors are derived as  $\frac{p}{p+q}$  and  $\frac{1}{q}$ , respectively. A packet is received correctly if the channel link remains in state *good* for the whole duration of packet transmission. Otherwise, it is received in error. Channel links between the AP and respective stations are independent of one another in their error characteristics. Note that this paper employs the two-state error model, as it can clearly address the advantage of the proposed scheme, but the scheme can be accommodated to the multi-state error model [15].

#### 3.2 Message Model

A station, currently inactive, can be activated by an upper layer query command that wants to monitor or process the data flow from the mobile station [6]. The query also specifies sampling period as well as precision level of needed data to determine the message length. In case of a change in the active flow set, the network schedule should be regenerated [9]. The destination of a message can be either within a cell or outside a cell, and the outbound messages are sent to the AP first and then forwarded to the remote destination. Hence, the AP also acts as another source of message stream. In addition, even

for the point-to-point streaming, multicast mode transmitting is assumed to deactivate the mandatory ACK, which may uncontrollably prolong the transmission procedure.

The sensory data is typically *isochronous* (or synchronous), consisting of message streams that are generated by their sources on a continuing basis and delivered to their respective destinations also on a continuing basis [9]. This paper follows the general real-time message model which has  $n$  streams, namely,  $S_1, S_2, \dots, S_n$ , and each  $S_i$  generates a message not more than  $C_i$  at each beginning of its period  $P_i$ . Each packet must be delivered to its destination within  $D_i$  from its generation at the source, otherwise, the packet is considered to be lost.  $D_i$  usually coincides with  $P_i$  to ensure that the transmission completes before the generation of the next message. Finally, each stream belongs to a specific station, so if a slot is assigned to a stream, it means that AP should poll that station.

## 4. DUAL CHANNEL OPERATION

### 4.1 Bandwidth Management

The network time is divided into a series of fixed-size slots, and each of them is exclusively assigned to a real-time station for the sake of real-time guarantee. The slot length, say  $L$ , is as large as the basic unit of wireless data transmission and all traffic is segmented to fit the slot size. On WLAN, such network access can be implemented by making AP poll each station according to the predefined schedule during the CFP, while each station transmits for as long as  $L$ . Even in the multi-hop network, a node can be synchronized by means of GPS (Global Positioning System) timing signals. To describe the goal of slot assignment, let  $\langle f_i^1, f_i^2 \rangle$  be  $i$ th slots of channel 1 and channel 2. If  $f_i^1$  and  $f_i^2$  are allocated to different streams, say  $A$  and  $B$ , respectively, switching transmission channels does not violate time constraints. Hence, we define  $\langle f_i^1, f_i^2 \rangle$  as a *switchable pair* if  $f_i^1$  and  $f_i^2$  are allocated to different streams, or any one of  $f_i^1$  and  $f_i^2$  is left unassigned. The purpose of bandwidth allocation, or slot assignment is to maximize the number of switchable pairs. For the AP to decide the polling order for a stream set, each stream  $S_i$  submits tuple of  $(P_i, C_i)$  information to the AP at connection set-up, where every  $P_i$  as well as  $C_i$  is an integer multiple of  $L$ . The bandwidth allocation consists of 3 steps, namely, stream partition, EDF based reservation, and slot rearrangement.

Step 1 begins with the schedulability test based on the following condition [9]:

$$\sum_{i=1}^n \frac{C_i}{P_i} + \Delta \leq 2.0,$$

which assumes that all messages from  $n$  streams are sorted by increasing relative deadline, while  $\Delta$  denotes the overhead term originated from the network management such as polling/probing overhead, beacon frame broadcast, interframe space, and so on. As  $\Delta$  is fixed for a stream set, we can calculate per-slot overhead, and merge it into  $C_i$ . Namely, if we let  $\Delta_p$  the per-slot overhead, the modified transmission time,  $C_i^{t+1}$ , can be calculated iteratively with the following equation until  $C_i^{t+1}$  becomes equal to  $C_i^t$ .

$$C_i^{t+1} = C_i^t + \left\lceil \frac{C_i^t}{L} \right\rceil \Delta_p$$

Hence,  $\Delta$  can be assumed to be 0, and we can concentrate on the problem of slot allocation. To inherit the efficiency of EDF in a single resource system, the allocation scheme first partitions the given stream set into two identical ones, each of which has the same period but its transmission time of every stream is reduced by half. Namely,

$$\Theta : \{(P_i, C_i)\} \rightarrow \Theta_1 : \left\{ \left( P_i, \frac{C_i}{2} \right) \right\}, \quad \Theta_2 : \left\{ \left( P_i, \frac{C_i}{2} \right) \right\}.$$

Next at step 2, the transmission is reserved for each stream. The AP should create a polling table for the entire polling sequence. For  $S_i$ , every message arrival is estimated at  $0, P_i, 2P_i$ , and so on, while each invocation needs  $\left\lceil \frac{C_i}{L} \right\rceil$  slots. Based on this estimation, the scheduling is virtually performed at each slot boundary to build a polling order table. The scheduler selects the message with the closest deadline and allocates the next slot to it. If no slot request is pending at a slot boundary, the scheduler leaves the slot unassigned. As the schedules for  $\Theta_1$  and  $\Theta_2$  are determined by EDF policy, both schedules are identical. Since the invocation behavior for a set of periodic tasks repeats itself once every  $T$  time units, where  $T$ , called the *planning cycle* of the task set, is the least common multiple of the periods of all tasks, we only need to consider task invocation in a planning cycle. In addition, a modification of some period makes it possible for the set to have a good planning cycle without violating the given timeliness requirement [16].

Finally in step 3, the allocation in  $\Theta_2$  is rearranged to maximize the number of switchable pairs. The current assignee of a slot has movable range. The earliest time of movement,  $E_t$ , is the arrival time of assignee of slot  $t$ , while the bound of backward movement is its deadline,  $L_t$ . The  $E_t$  and  $L_t$  of unassigned slot are set to 0 and  $T$ , respectively, as it can be reallocated anywhere within the planning cycle. From the last slot,  $f_t^1$  and  $f_t^2$  are investigated whether they are equal, namely, they are allocated to the same station. If so, the rearrangement procedure attempts to change  $f_t^2$  as follows:

```

for slot  $t$  from  $T - 1$  down to 0
  for slot  $i$  from  $E_t$  to  $L_t - 1$ 
    if ( $f_i^2 = f_t^2$ ) continue;           // same node
    if (not ( $E_t \leq i < L_t$ )) continue;
    if (not ( $E_i \leq t < L_i$ )) continue;
     $f_i^2 \leftrightarrow f_t^2$ ;  $E_i \leftrightarrow E_t$ ;  $L_i \leftrightarrow L_t$ ;
    break;
  end for
end for

```

**Example:** The example stream set consists of 5 streams, A(6, 2), B(4, 2), C(12, 2), D(3, 2), and E(8, 2). Their utilization is 1.92, while the planning cycle to 24. At step 1, the given stream set is partitioned into  $\Theta_1$  and  $\Theta_2$ , each of which is  $\{(6, 1), (4, 1), (12, 1), (3, 1), (8, 1)\}$ . Fig. 3 (a) shows the network schedule generated by step 1 in addition to  $E_t$

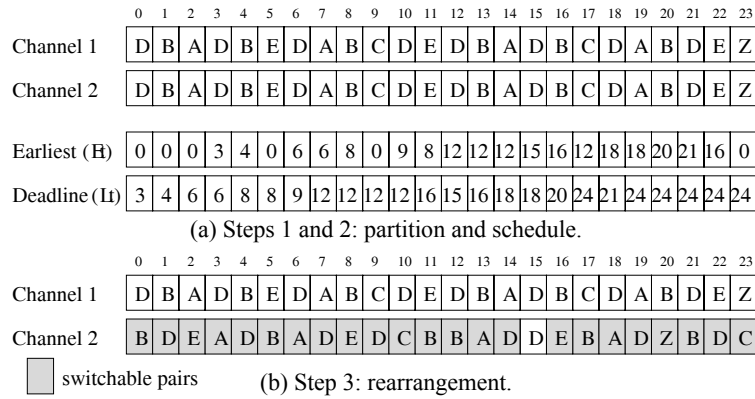


Fig. 3. Example of scheduling procedure.

and  $L_t$  of each slot. The rearrangement procedure starts from slot 23 backward to slot 0. In Fig. 3 (a), for  $t = 11$ ,  $f_{11}^1$  and  $f_{11}^2$  are both  $E$ , so it is desirable to relocate  $E$  in  $f_{11}^2$ . Among slots from 8 (decided by  $E_{11}$ ) to 15 (decided by  $L_{11}$ ), slot 8 is selected as  $f_8^2$  is  $B$  different from  $E$ ,  $E_{11} \leq 8 \leq L_{11}$ , and  $E_8 \leq 11 \leq L_8$ . Finally,  $f_8^2$  and  $f_{11}^2$  are exchanged, making  $\langle f_{11}^1, f_{11}^2 \rangle$  a switchable pair. Fig. 3 (b) shows the final allocation. In this example, 23 out of 24 slot pairs turned into the switchable pair.

## 4.2 Runtime Scheduling

In each slot, before polling a station, the AP transmits a probing control frame to the scheduled station, which then returns the control frame to the AP [5]. If the AP does not receive the probing control frame correctly from the station, the channel is estimated to be bad. Even though the probing indicates the channel is good, the ensuing transmission can fail if a state transits to a bad state during transmission. Let's assume that the AP is to start slot  $i$ , which is originally allocated to  $A$  on channel 1 as well as  $B$  on channel 2, namely,  $\langle A, B \rangle$ . The AP first probes the channel condition from itself to  $A$  and  $B$  on both channels, so each slot should inevitably contain two probing latencies. In case predicted channel conditions are available to the AP [15], probing can be replaced with the specific channel predication method.

During the runtime, the AP polls each stream according to the following 3 levels: At the first level, the AP selects the stream according to the static schedule decided at the bandwidth allocation step. When the channel estimation detects that some scheduled station is unreachable, the second level procedure switches polls as shown in Table 1. Finally, if even the channel switch cannot find an appropriate stream, the slot is given to another stream. Table 1 shows the probing result and corresponding actions. As shown in row 1, the AP can reach  $A$  on channel 1 and also  $B$  on channel 2, each node is polled as scheduled. Likewise, the channel switch is needless when all channels are in a bad state in row 11. However, in row 2, all connections from the AP are in a good state, except the connection to  $B$  through channel 2. In this case, only  $A$  can send on channel 1. However, if we switch  $\langle A, B \rangle$  to  $\langle B, A \rangle$ , both streams can successfully send their messages.

**Table 1. Channel status and transmission.**

| No. | Ch1-A | Ch2-B | Ch1-B | Ch2-A | Ch1 | Ch2 | save |
|-----|-------|-------|-------|-------|-----|-----|------|
| 1   | Good  | Good  | X     | X     | A   | B   | 0    |
| 2   | Good  | Bad   | Good  | Good  | B   | A   | 1    |
| 3   | Good  | Bad   | Good  | Bad   | A   | -   | 0    |
| 4   | Good  | Bad   | Bad   | X     | A   | -   | 0    |
| 5   | Bad   | Good  | Good  | Good  | B   | A   | 1    |
| 6   | Bad   | Good  | Good  | Bad   | -   | B   | 0    |
| 7   | Bad   | Good  | Bad   | X     | -   | B   | 0    |
| 8   | Bad   | Bad   | Good  | Good  | B   | A   | 2    |
| 9   | Bad   | Bad   | Good  | Bad   | B   | -   | 1    |
| 10  | Bad   | Bad   | Bad   | Good  | -   | A   | 1    |
| 11  | Bad   | Bad   | Bad   | Bad   | -   | -   | 0    |

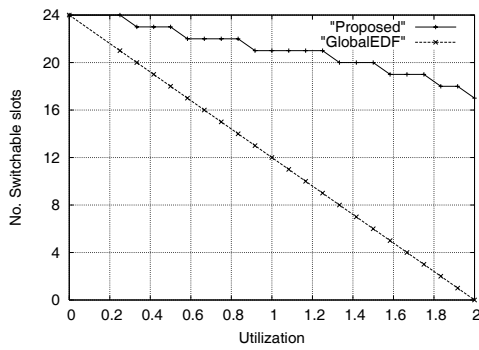
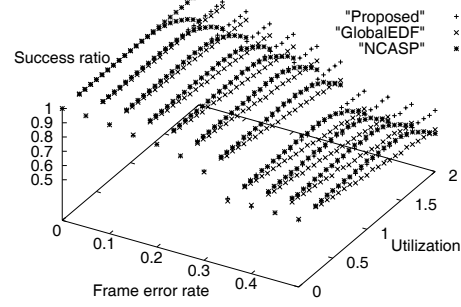
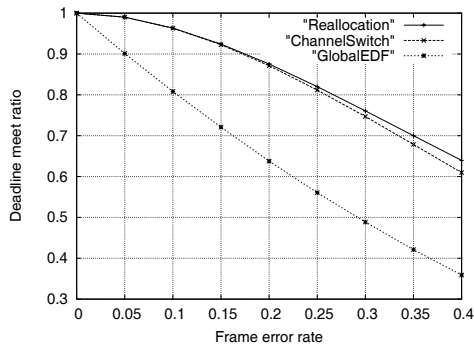
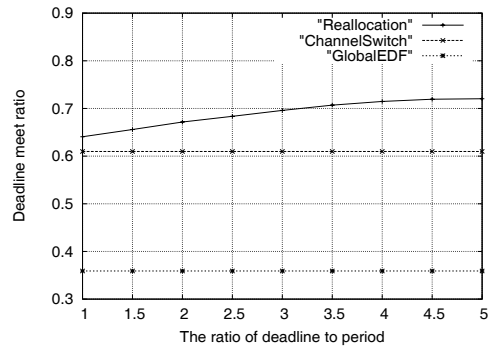
We can save one transmission loss, so its *save* column is marked as 1. Row 8 describes the situation that the AP can reversely reach *A* only on channel 2 while *B* on channel 1. By switching polls between the two channels, we can save the 2 transmissions that might fail on the ordinary schedule.

Table 1 has some entries marked as ‘-’; which means the AP can poll neither *A* nor *B* at the slot. In this case, it seems better to poll another stream, making the original stream be polled at the next time. To this end, the AP maintains a queue for deferred messages ordered by their deadlines. A deferred poll enters the queue to be fetched if the AP has nothing to poll. When the deadline of a message expires, the entry will be automatically eliminated from the queue. In case this queue is also empty, the AP may poll in advance a stream that is already pending. The successful transmission of such a slot can make the next slot unassigned, giving more chances to recover from other transmission errors.

## 5. PERFORMANCE EVALUATION

This section measures the performance of the proposed scheme in terms of switchable pairs and corresponding success ratio according to the frame error rate via simulation using the ns-2 event scheduler [17]. For the first experiment, we fixed the length of planning cycle to 24 as well as the number of streams to 3, and generated every possible stream set with utilization ranges from 0.2 to 2.0 in order to measure how the rearrangement scheme can generate switchable pairs. Fig. 4 plots the measurement result sorted by utilization of stream sets. Even when utilization is 2.0, on average, 17 out of 24 slots are rearranged as switchable pairs. As utilization becomes lower, the number of switchable pairs increases, since the number of unassigned slots also grows. Global EDF generates the switchable pair mainly by unassigned slots, so the number of switchable pairs proportionally decreases as utilization increases. The gap between the two schemes becomes larger under higher utilization. When the number of streams exceeds 3 with planning cycle fixed to 24, almost every slot becomes switchable.

Fig. 5 plots the success ratio according to the frame error rate along with utilization, to demonstrate the effect of slot rearrangement. The success ratio means the ratio of


 Fig. 4. Number of switchable slots ( $n = 3$ ).

 Fig. 5. Success ratio ( $n = 3$ ).

 Fig. 6. The impact of error rate ( $n = 5$ ).

 Fig. 7. The impact of deadline ( $n = 5$ ).

timely delivered real-time messages to all generated messages. The frame error rate is a function of frame length and bit error rate, and is set to range from 0.0 to 0.4, considering the error-prone nature of wireless channels. This experiment compared the success ratio of the proposed scheme with those of global EDF and NCASP. The value of  $\phi$  is ideally chosen to be 12. As shown in Fig. 5, first of all, when the frame error rate is 0.0, every transmission succeeds in both our scheme and global EDF regardless of utilization. However, NCASP misses some deadlines because it cannot optimally schedule real-time packets when utilization exceeds 1.3. The proposed scheme outperforms the global EDF by around 18% at maximum due to differences in the number of switchable pairs when the frame error rate becomes closer to 0.4. The performance gap becomes larger according to the increase in frame error rate. On lower utilization, NCASP shows almost same performance as the proposed scheme since it assigns a series of slots to a stream, increasing the number of switchable pairs. However, the limitation of real-time scheduling capability causes performance degradation under high utilization.

Next, experiments are performed to measure the effect of respected components. In this case, to show the efficiency of the reallocation, the number of streams is set to 5. When the channel switch cannot find the stream to poll, the AP picks a stream among the remaining 3 streams. We also generated every possible stream set of which utilization has the value of 1.6 through 1.8. Fig. 6 plots the deadline meet ratios of the proposed scheme and Global EDF according to the frame error rate. In this figure, the *Channel-*

*Switch* curve shows the effect of channel switch across the two networks, while the curve *Reallocation* shows the effect of dynamic error recovery. The channel switch can enhance the success ratio up to 24.1% while the reallocation brings an additional 3.9% improvement, showing the capability to cope with severe channel errors. It can be expected that the success ratio can be improved according to the increase of the number of stream sets, as there are many candidates to replace the original schedule. However, when the set has more than 5 streams, the reallocation step has enough candidates, so the influence is negligible.

Fig. 7 exhibits the success ratios according to the length of deadline. The deadline of each stream is decided by the ratio to its period, while the ratio has the value of 1.0 through 5.0. The frame error ratio is set to 0.4 to emphasize the effect of the deadline. Needless to say, the larger the deadline, the more time is left for error control. The curves for *ChannelSwitch* and *GlobalEDF* remain almost constant, as they lack the functionality of dynamic reallocation. However, the *Reallocation* can enhance the deadline meet ratio by up to 11%.

## 6. CONCLUSION

This paper has proposed and analyzed the performance of real-time message scheduling policy for dual channel wireless sensor networks. The proposed scheme coped with NP-hard complexity of multiple-resource scheduling by evenly partitioning the stream set into two identical sets to apply the EDF policy that is optimal under uniprocessor environments for dynamic preemptive real-time scheduling. In addition, slot rearrangement maximizes the number of switchable pairs, so the AP can dynamically select the error-free channel for two streams. In addition, the channel reallocation step with the support of the deferred message queue enables a message to be retransmitted during the slots unreachable to the scheduled streams. Simulation results show that the proposed scheme can generate 70% of switchable pairs even when the utilization reaches the saturation point, improving the ratio of timely delivery by up to 18% when the frame error rate approaches 0.4. In addition, the slot reallocation further enhances real-time performance up to 3.9%, when the frame error rate is around 0.4 and up to 11% when the deadline of each stream more than 5 times as long as its period.

Dual channels are exploited just to enhance the timeliness of real-time message transmissions and the proposed scheme can be seamlessly extended to multiple channels. However, each channel can be scheduled to achieve multiple goals. For example, one channel schedules according to EDF, while the other channel selects the stream which has been most degraded to reflect the fairness issue. Then dynamic channel switch and channel reallocation enable us to pursue respective goals without sacrificing one for the other. In addition, we are planning to design an efficient scheduling scheme for the dual channels that support multiple transmission rates rather than the Gilbert error model [15]. In this case, we believe that the guarantee for the real-time stream needs additional assumptions, for example, the stream meets its deadline when the minimum or average transmission rate is above a certain rate.

## REFERENCES

1. A. Patil and N. Audsley, "Implementing application specific RTOS policies using reflection," in *Proceedings of the 11th IEEE Real Time and Embedded Technology and Applications Symposium*, 2005, pp. 438-447.
2. N. Ignat, B. Nicolescu, Y. Savaria, and G. Nicolescu, "Soft-error classification and impact analysis on real-time operating systems," in *Proceedings of the Conference on Design, Automation and Test in Europe*, 2006, pp. 182-187.
3. H. Li, P. Shenoy, and K. Ramamritham, "Scheduling communication in real-time sensor applications," in *Proceedings of the 10th IEEE Real-Time Embedded Technology and Applications Symposium*, 2004, pp. 2065-2080.
4. E. Ngai, Y. Zhou, M. Lyu, and J. Lui, "Reliable reporting of delay-sensitive events in wireless sensor-actuator networks," in *Proceedings of IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, 2006, pp. 101-118.
5. S. Choi and K. Shin, "A unified wireless LAN architecture for real-time and non-real-time communication services," *IEEE/ACM Transactions on Networking*, Vol. 8, 2000, pp. 44-59.
6. M. Adamou, S. Khanna, I. Lee, I. Shin, and S. Zhou, "Fair real-time traffic scheduling over a wireless LAN," in *Proceedings of the 22nd IEEE Real-Time Systems Symposium*, 2001, pp. 279-288.
7. A. Rangnekar, C. Wang, K. Sivalingam, and B. Li, "Multiple access protocols and scheduling algorithms of multiple channel wireless networks," *Handbook of Algorithms for Mobile and Wireless Networking and Computing*, C.H.I.P.S., Texas, 2004.
8. J. Carpenter, S. Funk, P. Holman, A. Srinivasan, J. Anderson, and S. Baruah, "A categorization of real-time multiprocessor scheduling problems and algorithms," *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, Chapman & Hall/CRC Press, 2003.
9. J. Liu, *Real-Time Systems*, Prentice Hall, 2000.
10. IEEE Std 802.11-1999, Part 11-Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.
11. A. Raniwala and T. Chiueh, "Architecture and algorithms for an IEEE 802-11-based multi-channel wireless mesh network," in *Proceedings of IEEE INFOCOM*, 2005, pp. 2223-2234.
12. S. Damodaran and K. Sivalingam, "Scheduling algorithms for multiple channel wireless local area networks," *Computer Communications*, Vol. 25, 2002, pp. 1305-1314.
13. U. Devi, H. Leontyev, and J. Anderson, "Efficient synchronization under global EDF scheduling on multiprocessors," in *Proceedings of the 18th EUROMICRO Conference on Real-Time Systems*, 2006, pp. 128-138.
14. A. Khattab and K. Elsayed, "Channel-quality dependent earliest deadline due fair scheduling schemes for wireless multimedia networks," in *Proceedings of the 7th ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2004, pp. 31-38.
15. T. Carley, M. Ba, R. Barua, and D. Stewart, "Contention-free periodic message scheduler medium access control in wireless sensor/actuator networks," in *Proceedings of the 24th IEEE Real-Time Systems Symposium*, 2003, pp. 298-307.
16. K. Fall and K. Varadhan, "Ns notes and documentation," Technical Report, VINT project, UC-Berkeley and LBNL, 1997.



**Junghoon Lee (李政勳)** received the B.S., M.S., and Ph.D. degrees at Department of Computer Engineering, Seoul National University, Korea. From 1990 to 1992, and also in 1996, he was a senior research engineer at Lab. of Optical Telecommunication, Mercury, Korea. In 1997, he joined the Department of Computer Science and Statistics at Cheju National University, Korea where he is currently an associate professor. From 2003 to 2005, he was a visiting scholar at Department of Computer Science, University of Texas at Austin. His research interests include real-time communication and wireless network.



**Mikyung Kang (姜美京)** received the B.S., M.S., and Ph.D. degrees at Department of Computer Science and Statistics from Cheju National University, Jeju, Korea. From 2004 to 2006, she was a researcher at ITRC (Information Technology Research Center) of Cheju National University. In 2006, she was selected as a Post-Doctoral Fellow by KRF (Korea Research Foundation). Currently, she is a research scholar at University of Southern California's ISI-East (Information Sciences Institute East). Her research interests are real-time communication and wireless network.



**Gyung-Leen Park (朴景麟)** received B.S. in Department of Computer Science from Chung-Ang University. He received M.S. and Ph.D. from Computer Science and Engineering department at the University of Texas at Arlington, respectively. His research interests include scheduling in parallel and distributed systems, mobile computing, and telematics. In 1997, he was a visiting assistant professor at the University of Texas at Arlington. In 1998, he joined the Department of Computer Science and Statistics at Cheju National University, Korea, where he is currently an associate professor. Currently, he is the director of the ITRC (Information Technology Research Center) of Cheju National University, Korea.



**Sung Y. Shin** received his Ph.D. in Computer Science from University of Wyoming in 1991. Dr. Shin is a professor and graduate coordinator of EE & Computer Science department at South Dakota State University. He is currently working as an invited professor of Computer Science and Engineering at Seoul National University, Seoul, Korea. He worked as a visiting scientist in Space and Life Science Division at NASA, Johnson Space Center in Houston, Texas in 1999 and 2001. He organized

numerous international conferences, and he is currently general co-chair of ACM SAC 2007, Seoul, Korea. His research interests are GIS, Telemedicine and Software Fault Tolerance has been funded by NASA and NSF, and he has published over 100 technical papers.