Design of a prioritized error control scheme based on load differentiation for time sensitive traffic on the wireless LAN

Mikyung Kang, Junghoon Lee*, Yongmoon Jin, Gyung-Leen Park, Hanil Kim
Dept. of Computer Science and Statistics, Cheju National University,
690-756, Jeju Do, Republic of Korea
Email: {mkkang, jhlee, ymjin, glpark, hikim}@cheju.ac.kr

***Abstract*—This paper proposes and analyzes the performance of a prioritized error control scheme for time sensitive application on the wireless sensor network. As a modified version of IEEE 802.11 WLAN, the proposed scheme further divides DCF into H-DCF and L-DCF without changing PCF, aiming at maximizing the successful retransmission of a packet that carries critical data. While channel estimation elminates the unnecessary polls to the sensor node currently unreachable during PCF, two DCF subperiods enable prioritized error recovery by making only the high priority packet be retransmitted via H-DCF. A good chop value, which distributes the retransmission to each period, can maximize recovered weight, or criticality, minimizing the possible degradation of network throughput. The simulation results show that the proposed scheme can improve recovered weight by 8% while showing 97% successful transmission at maximum for the given simulation parameter.**

***Index Terms*—Time-sensitive application, wireless sensor network, IEEE 802.11 WLAN, Prioritized error control, Packet criticality**

I. INTRODUCTION

In the past few years, smart sensor devices have matured to the point that it is now feasible to deploy a large, distributed network of such sensors [1]. Sensor networks are differentiated from other wireless, battery-powered environments in that they consist of tens or hundreds of autonomous nodes that operate without human interactions for a long time. The sensor network makes possible a new range of application such as environmental monitoring and control. Wireless sensor/actuator networks allow scientists to monitor remote environmental conditions such as temperature, pressure, and chemical presence. Such monitoring network can be used to detect forest fires and alert authorities, or even extinguish the fire. Besides, the application of sensor networks is being expanded to industrial, military, agricultural, and other various areas.

Communication between the sensors and sinks requires a network [2]. Since sensor applications preclude the use of wired networks, wireless networks are commonly used in those applications. Wireless networks are inherently broadcast media, and all nodes in the network share one common communication medium. Therefore, a method for resolving contention when multiple nodes require access to the medium is necessary. This is the purpose of a MAC (Medium Access Control) protocol. The MAC protocol defines how and when nodes may access the medium. It must ensure that nodes share the medium in such a way that application requirements are met. Message flows exchanged in a sensor network are mainly periodic and need guaranteed delay for a computing node to make a meaningful and timely decision [3]. Many real-time scheduling and fair packet scheduling algorithms have been developed for wired networks, however, it is not clear how well these algorithms work for wireless sensor networks where channels are subject to unpredictable location-dependent and time-varying bursty errors [4]. The wireless link is highly variable even over short distances due to the statistical distribution of path loss and the physical properties of propagation environment. In the presence of such unpredictable errors, real-time traffic applications cannot fully utilize the channel bandwidth assigned to them.

The IEEE 802.11 was developed as a MAC standard for WLAN [5]. The standard consists of a basic DCF (Distributed Coordination Function) and an optional PCF (Point Coordination Function). The DCF exploits CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) protocol for non-real-time messages, aiming at enhancing their average delivery time as well as network utilization. However, packet collisions, intrinsic to CSMA protocols, make it impossible for a node to predictably access the network. After all, QoS guarantee cannot be provided without developing a deterministic access schedule on top of collision-free PCF, in which a point coordinator polls active sensor nodes [6]. The polling schedule is decided according to the contract negotiated between each flow and the point coordinator.

Based on "Design of an efficient error control scheme for time-sensitive application on the wireless sensor network based on IEEE 802.11 standard", by Junghoon Lee, Mikyung Kang, Yongmoon Jin, Gyungleen Park, and Hanil Kim which appeared in the Proceedings (LNCS) on IWDC 2005, Kharagpur, India, December 2005.

* Corresponding author

PCF thus makes WLAN as a powerful and promising infrastructure for the wireless sensor network of time-sensitive sensor application.

However, WLAN needs to efficiently deal with network error during the delivery of sensory data. First, it is desirable to skip polling a node whose channel is not in normal condition. Second, once the packet transmission fails, it should be retried in a best-effort manner within its deadline. Third, due to the bursty nature of wireless network error, not all packets can be recovered by retransmission. In that case, the retransmission should carefully consider the priority of a packet, and network should try to enhance the successful retransmission of higher priority packets. The priority of packet can be decided by the importance of data the packet carries, or by the degree of QoS degradation the flow experienced [4]. If we consider a monitoring application, a momentary event data is more important than other steady state data. The bandwidth for the respective error control packet cannot be reserved for each flow during PCF, for it will lead to the great waste of network bandwidth and inefficiency. In addition, it is impossible for AP to know which node wants to be polled for retransmission on a specific time. Hence, it is natural that error control is performed on DCF, making sensor nodes fairly share the bandwidth.

Implementing a global priority resolution in DCF seems to be unrealistic as it needs major modification on the original CSMA/CA protocol as well as priority resolving procedure for every packet transmission invokes so much overhead that can be avoided when there is no contending message [7]. In this paper, we propose and analyze an error control scheme for sensor data on DCF interval of IEEE 802.11 WLAN. It is aiming at supporting, though limited, level of priority in recovering the packet transmission error, during DCF without greatly modifying the original CSMA/CA protocol. To this end, AP divides the DCF into two subperiods, makes their loads different, and gives more chance to the higher priority message by transmitting it via lower load network. This policy enhances the possibility of successful transmission for higher priority packets, insignificantly sacrificing the overall network throughput.

The rest of this paper is organized as follows: Section II introduces the background of this paper, including the related works on real-time communications on the wireless medium, IEEE 802.11 Wireless LAN standards, and real-time message model. Then Section III proposes the communication architecture for time-sensitive sensor traffic, describing channel estimation, bandwidth allocation, and prioritized error control. After demonstrating the simulation result in Section IV, Section V concludes this paper with a brief summarization and the description of future works.

II. BACKGROUND

A. Related Works

As an example of pure bandwidth allocation scheme, DBASE (Distributed Bandwidth Allocation/ Sharing/ Extension) is a protocol capable of supporting both synchronous and multimedia traffics over IEEE 802.11 *ad hoc* WLAN where no fixed access point coordinates medium access [8]. The basic concept is that each time real-time station transmits its packet it will also declare and reserve the needed bandwidth at the next CFP. Every station collects this information and then calculates its actual bandwidth at the next cycle. But it focuses on deciding whether to accept the respective real-time packet without considering a flow-level allocation or negotiation.

M. Caccamo and et. al. have proposed a MAC that supports deterministic real-time scheduling via the implementation of TDMA (Time Division Multiple Access), in which the time axis is divided into fixed size slots [6]. 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. Each message implicitly contends for the medium through the scheduling algorithm, for example with priorities, instead of explicitly on the medium. Unfortunately, to implement implicit contention, each node must schedule all messages in the network, resulting in the complexity growing linearly with the number of messages. In addition, their scheme didn't consider the network error at all, just focusing on the scheduling policy.

Choi and Shin suggested a unified protocol for real-time and non-real-time communications in wireless networks [3]. In their scheme, a BS (Base Station) polls a real-time mobile station according to the non-preemptive EDF (Earliest Deadline First) policy. The BS also polls the non-real-time message according to the modified round-robin scheme regardless of a standard CSMA/CA protocol to completely eliminate a message collision. The retransmission of a damaged packet is considered as a normal non-real-time message. Additionally, to handle location-dependent, time-varying, and bursty channel errors, the channel state can be predicted via channel probing before the packet is transmitted. That is, before transmitting a downlink packet to the mobile, the BS transmits a probing control packet to that mobile, which then returns the control packet to the BS. With this estimation, it is possible to reduce the need for retransmission, since retransmission can be harmful in meeting deadlines.

Adamou and his colleagues have addressed the scheduling problem of achieving fairness among real-time flows with deadline constraints as well as maximizing the throughput of all the real-time flows over a wireless LAN [4]. They chose the scheduling objective of minimizing the maximum degree of the degraded QoS among all applications. Their scheduling policy includes EDF (Earliest Deadline First), GDF (Greatest Degradation First), EOG (EDF Or GDF), and LFF (Lagging Flows First). The BS performs the scheduling of real-time packet deliveries using a polling scheme. This scheme is built on the assumption that BS knows which station has messages to retransmit as well as their deadlines, and decides which one to poll among them according to the criteria described above.

In addition to the retransmission technique, there exists a way the sender adjusts the transmission rate according to the currently available network bandwidth. Shah proposed a dynamic bandwidth management scheme in a single-hop ad hoc wireless network that is especially suitable for hot-spot networks, that is, a number of nodes in a small area share limited channel bandwidth [9]. As the available channel capacity changes and also the traffic characteristics of various flows change, the bandwidth manager dynamically reallocates the channel access time to the individual flows.

When network contention and data rates are low due to the poor network condition, the transmission queue can be drained faster than sampling results arrive [1]. However, because the number of messages produced during a single epoch can vary dramatically, there are situations when the queue will overflow. In this case, the system must decide which one to discard. In the *naive* scheme, no data is considered more valuable than any other, so the queue is drained in a FIFO manner and data are dropped if they do not fit in the queue. The *winavg* scheme works similarly, except that instead of dropping results when the queue fills, the two results at the head of the queue are averaged to make room for the new result. Since the head of the queue is now an average of multiple records, we associate a count with it. In the *delta* scheme, a data value is assigned an initial score relative to its difference from the most recent value successfully transmitted from this node, and at each point in time, the data with the highest score will be delivered.

B. IEEE 802.11 WLAN

Two different key approaches can be followed in the implementation of a WLAN: an *infrastructure-based* approach and an *ad-hoc networking* one. The infrastructure-based architecture imposes the existence of a centralized controller for each cell, often referred to as AP (Access Point). The AP is normally a gateway to the external networks, for example, wired backbone network, or other wireless networks. Whereas, an ad-hoc network is a peer-to-peer network formed by a set of stations within the range of each other that dynamically configure themselves to set up a temporary network. PCF access cannot be adopted in ad-hoc networks.

The wireless LAN operates on both CP (Collision Period) and CFP (Collision Free Period) phases alternatively in BSS (Basic Service Set) as shown in Fig. 1. Each superframe consists of CFP and CP, which are mapped to PCF and DCF, respectively. PC (Point Coordinator) node, typically AP, sequentially polls each station during CFP. Even in the *ad hoc* mode, it is possible to designate a specific node to play a role of PC in a target group. Only the polled node is given the right to transmit its message for a predefined time interval, and it always responds to a poll immediately whether it has a pending message or not. In contrast, DCF is the basis of the standard CSMA/CA access mechanism and it uses the RTS (Request To Send)/CTS (Clear To Send) clearing technique to further reduce the possibility of collisions. After sending a RTS, subsequent steps take place using SIFS (Short InterFrame Space), so the transmission of a frame is an atomic operation that cannot be interrupted by another packet.

The phase of network operation is managed by the exchange of control packets which have higher priority than other packets. The prioritized access is achieved by different length of IFS the node waits before it attempts to send its packet. The PC attempts to initiate CFP by broadcasting a *Beacon* at regular intervals derived from a network parameter of *CFPRate*. Round robin is one of the popular polling policies for CFP, in which every node is polled once a polling round. A polling round may be completed within one superframe, or spreads over more than one superframe. In case the CFP terminates before all stations have been completely polled, the polling list is resumed at the next node in the ensuing CFP cycle.

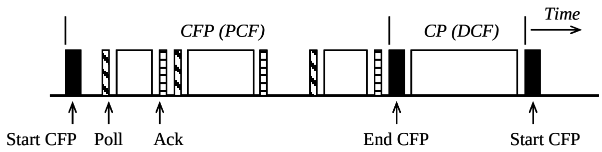

Figure 1. Time axis of wireless LAN

Fig. 1 also illustrates that poll, transmission, and acknowledgment are atomic, namely, these steps must complete in their entirety to be successful. Senders expect acknowledgment for each transmitted frame and are responsible for retrying the transmission. After all, error detection and recovery is up to the sender station, as positive acknowledgments are the only indication of success. If an acknowledgment is expected but does not arrive, the sender considers the transmission failed.

C. Message Model

There is no other traffic but the sensor data in the network. The traffic of sensored 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 [10]. A node, currently inactive, can be activated by an upper layer query command that wants to monitor or process the data flow from the sensor node. The destination of message can be either within a cell or outside a cell, and the outbound messages are first sent to the router node such as AP and then forwarded to the final destination. Internal messages are also relayed by the AP. The query may also specify sampling period and the precision level of needed data (hence, message length) on a node. In case of a change in the active flow set, bandwidth is to be reallocated or network schedule mode is changed [9]. This paper follows the general real-time message model which has n streams, namely, S_1, S_2, ..., S_n, and for each S_i, a message sized less than C_i is produced at the beginning of its period, P_i. Each packet must be delivered to its destination within D_i units of time from its generation or arrival at the source, otherwise, the packet is considered to be lost. We assume that D_i is larger than P_i to give a sufficient margin to recover transmission failures. When S_i is polled, it can transmit up to H_i, and $\{H_i\}$ is named as *capacity vector*. The sampled data has its own weight and the weight should be mapped to the corresponding priority level.

III. Message Scheduling Scheme

A. Overview of Network Operation

The entire network consists of a number of autonomous cells and each of them has its own AP to manage the message schedule. After a query issued to the entire sensor network is analyzed and disseminated, each subquery is assigned to respective network to activate a necessary sensor operation. Hence, we focus on the bandwidth allocation and error control within a cell. According to the operation of AP, the time axis of WLAN consists of a series of superframes and each of them consists of PCF, H-DCF, and L-DCF, virtually dividing the link into 3 discrete channels. As previously mentioned, the beacon frame initiates each phase one by one.

Naturally, each channel can interfere with one another, due to the deferred beacon problem, that is, a beacon message can get delayed and the start of PCF can be put off, if another packet is already occupying the network. This is because only after the medium is idle the coordinator will get the priority through the shorter IFS. The maximum amount of deferment coincides with the maximum length of a data packet, as can be inferred in Fig. 2. This interference may jeopardize the sophisticated network schedule that guarantees satisfying QoS requirement, particularly in case the deadline of message is equal to its period. Otherwise, the delayed start of PCF just induces a delay jitter that can be absorbed by the receiver. Additionally, we assume that the length of PCF and that of H-DCF are not reduced even if their starts are delayed. Just L-DCF shrinks its length when its start gets delayed, as shown in the right-hand part of Fig. 2. After all, the time partition is equivalent to the three independent links.

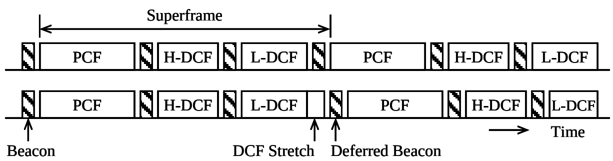

Figure 2. Time axis of proposed network

Guaranteed flows occupy the PCF under the exclusive control of AP. Each node transmits its message on each poll for the predefined time duration decided by a specific bandwidth allocation scheme. AP polls only those nodes whose channel is estimated to be *good*. Hence, during PCF, AP does not poll a node if its channel status is not estimated as good, since it has almost no possibility to success considering the error characteristics of wireless channel. If a transmission fails or is deferred, the sender moves the packet to the retransmission queue, as its importance is not as high as newly generated sensor data. The packet may have a chance to be retransmitted at that queue via H-DCF or L-DCF according to its priority.

B. Channel Estimation

The 802.11 radio channel is modeled as a Gilbert channel [11]. 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. The pair of p and q representing a range of channel conditions, has been obtained by using the trace-based channel estimation. The average error probability, denoted by ε, and the average length of a burst of errors are derived as $\frac{p}{p+q}$ and $\frac{1}{q}$, respectively. The packet is received correctly if the channel is in state *good* for the whole duration of packet transmission, otherwise, it is received in error. In case of an unsuccessful transmission, the station may retransmit the frame through either another CP or CFP according to the control policy.

We take the estimation method from Bottigliengo's work [12]. To trace the channel status, AP maintains a state machine, or simply flag, associated to each sensor node. The state can be either *good* or *bad* according to the channel condition. The channel condition is estimated as follows: The ACK/NAK is sent from the receiver to AP as soon as it receives a packet. If the AP does not receive an ACK/NAK within predefined time-out interval, the packet will be assumed to be lost. AP sets the state to *good* whenever it receives from the corresponding sensor node, namely, a MAC-layer acknowledgment in response to a data frame, a CTS frame in response to an RTS frame, or any other error-free frame. The AP sets the state to *bad* after a transmission failure. Each *bad* channel has its own counter, and when a counter expires the AP attempts to send a single data frame to check the channel status. The duration of timer is reset to its initial value upon a transmission from *bad* to *good*, and the value is doubled whenever the probing fails in *bad* state. The value of timer should be set small so as to quickly recover from short channel error period.

C. Bandwidth Allocation

By allocation, we mean the procedure of determining capacity vector, $\{H_i\}$, for the given superframe time, F, as well as message stream set, $\{S_i\,(P_i, C_i)\}$. This step is necessary for the network to guarantee the timely delivery of data packet generated by currently active sensors. For each change in the set of active sensors or their traffic characteristics, the polling schedule and capacity vector should be reallocated. The sampling period and the message length are different for each sensor, or they can be assumed to be tunable. There have been a lot of bandwidth allocation schemes for the real-time message stream or sensor data stream, we begin with Lee's scheme for its message model coincides with that of this paper. However, any other bandwidth allocation scheme can be applied with a trivial modification of their basic assumption or network model [13].

It is desirable that the superframe time is a hyperperiod of each stream's period and it is known that a message set can be made harmonic by reducing some periods by at most half [2]. So we assume that the superframe time is also given in advance, focusing on the determination of capacity vector. H_i limits the maximum time amount for which S_i can send its message. Let δ denote the total overhead of a superframe including polling latency, IFS and the like, while D_{max} the maximum length of a data packet. For a minimal requirement, F should be sufficiently large enough to make the polling overhead insignificant. In addition, if P_{min} is the smallest element of set P_i, F should be less than P_{min} so that every stream can meet at least one superframe within its period. After all, the requirement for the superframe time, F, can be summarized as follows:

$$\sum H_i + \delta + D_{\max} \le F \le P_{\min} \qquad (1)$$

In addition, the minimum value of available transmission time, X_i is calculated as in (2). Namely,

$$\begin{aligned} X_i &= \left(\left\lfloor \tfrac{P_i}{F} \right\rfloor - 1\right) \cdot H_i \quad && if \left(P_i - \left\lfloor \tfrac{P_i}{F} \right\rfloor \cdot F\right) \le D_{\max} \\ X_i &= \left\lfloor \tfrac{P_i}{F} \right\rfloor \cdot H_i && Otherwise \end{aligned} \qquad (2)$$

For each message stream, X_i should be greater than or equal to C_i ($X_i \ge C_i$). By substituting (2) for this inequality, we can obtain the least bound of H_i that can meet the time constraint of S_i.

$$\begin{aligned} H_i &= \frac{C_i}{\left(\left\lfloor \frac{P_i}{F} \right\rfloor - 1\right)} \quad && if \left(P_i - \left\lfloor \tfrac{P_i}{F} \right\rfloor \cdot F\right) \le D_{\max} \\ H_i &= \frac{C_i}{\left\lfloor \frac{P_i}{F} \right\rfloor} && Otherwise \end{aligned} \qquad (3)$$

The allocation vector calculated by (3) is a feasible schedule if it satisfies (1). By this, we can determine the length of CFP period (T_{CFP}) and that of CP (T_{CP}) as follows:

$$T_{CFP} = \sum H_i + \delta,\ T_{CP} = F - T_{CFP} \ge D_{\max} \qquad (4)$$

D. Scheduling of Retransmission

As shown in Fig. 3, each sensor node has two separate queues, namely, *normal* queue for ordinary packets and *retransmission* queue for the packets that are to be resent. Packets are ordered according to their priority in the retransmission queue, and a packet entry is automatically dropped when its deadline expires. This figure also illustrates that the proposed system has 3 virtual transmission links, PCF link, high-priority DCF link, and low-priority DCF link, while each of them is mapped to PCF, H-DCF, and L-DCF periods, respectively.

If a channel between AP and a node has been in a *bad* state for not a short duration, the number of packets to retransmit also increases. In that case, not all of packets can be recovered via retransmission due to the limit of available bandwidth. So it is necessary to give packets a certain type of priority and the network should give the precedence to the higher priority packets. The priority is typically decided by the importance of the value, or the degree of quality degradation for a flow.

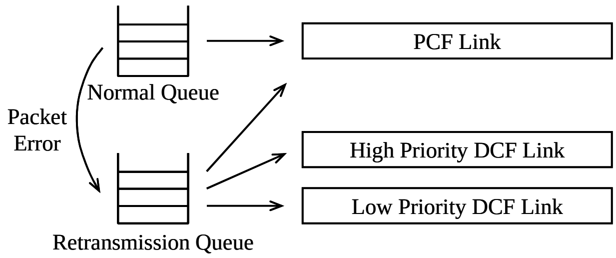

Figure 3. Queue Discipline

The regular packets are transmitted via PCF link when AP polls a node, and the polling schedule is decided by the bandwidth allocation scheme. In case AP skips some nodes as they are in *bad* state, other nodes can be polled more than guaranteed. Besides the node possibly has no packet in its normal queue, then it sends a packet in the retransmission queue, if any. The operation of H-DCF and L-DCF is as follows: The lower the load, the higher the probability of successful transmission. Hence, we are to make the load of H-DCF lower than that of L-DCF, actually differentiating the upper bounds of maximum load for two periods. However, as there is no global view for each node, they cannot know whether other nodes have higher priority packets or not. Consequently, H-DCF transmits those packets whose priority is higher than c, as shown in Fig. 4. Namely, they do not permit low priority packet to be resent during H-DCF, even if there is no traffic in H-DCF. It is possible for some packets to collide due to the same backoff value. Then the transmission will be retried after another backoff procedure just within one H-DCF. If a packet recovery fails in H-DCF, it can be retried in the L-DCF with a normal CSMA/CA procedure.

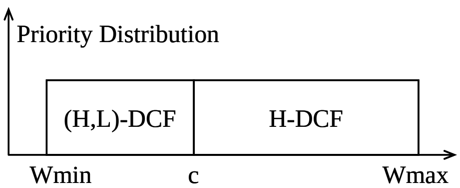

Figure 4. Chop Partition

The value c is a tunable parameter that can be set according to the network load, current error rate, weight distribution, and so on [14]. It ranges from the lowest priority value, W_{min} to the highest one, W_{max}. If c is set to W_{max}, it does not use the prioritized retransmission. The optimal value of c which maximizes value of recovered weight, can be found empirically or via analytical model for the given network parameters. Otherwise, some functions like bandwidth management may dynamically adjust the chop value. However, if a value exists and once a value is found, it can be used or adjusted according to the change of network parameter. Hence it is our concern to confirm that such values exist. It is natural that the number of recovered packet with non-partitioned scheme is obviously larger than that with the proposed partitioning scheme, as the latter leaves the H-DCF vacant to always reserve bandwidth for the higher priority packets, resulting in declining network throughput.

IV. Performance analysis

At first, we considered ns-2 which is the most common simulation tool for wireless network, but it lacks the support for direct link layer control or error simulation model [15]. Thus the experiments are fulfilled via SMPL [16], which is functionally equivalent to ns-2 event scheduler. With SMPL, we implemented restricted contention protocol based on RTS/CTS mechanism for DCF. To concentrate on the prioritized error recovery in wireless sensor networks, we simplified the experiment as follows: First of all, all time variables are aligned to the length of superframe time, F. Every stream has equal period, message length, and deadline, and F exactly divides P_i while the deadline is fixed to $5F$. Each packet is generated at constant intervals without any jitter and sent as a frame. The number of active sensors is 5 and their utilization is 0.5, while capacity vector is $\{0.1F, 0.1F, ..., 0.1F\}$. Each packet fits to the length of $0.1F$, being associated to a priority randomly picked from 0 to 19. We also, rather unrealistically, assume that error estimation is always correct.

The first experiment measures the effect of chop value with fixed error rate, ε. The ε is set to 0.01, while the length of error duration, denoted as $\frac{1}{q}$ in Gilbert error model, distributes exponentially with average $2.0F$. The y-axis of Fig. 5 is the ratio of total weights of recovered packets to those of packets that need retransmission. As natural, some packets are discarded at the retransmission queue due to deadline expiration. As shown in Fig. 5, the chop value is critical to the overall performance. This figure also plots the recovered weights by retransmission via non-partitioned DCF through ordinary CSMA/CA protocol to compare with the proposed scheme. The gap between the two curves is maximized when chop value is 0.55. After that point, the larger is chop value, the more bandwidth is wasted. As contrast, with smaller chop value, most of packets are attempted on H-DCF and again on L-DCF, increasing the possibility of collision.

Fig. 6 plots the measurement result of recovered weights according to the ε ranging from 10^{-3} to 10^{-2}. For each ε value, each experiment runs with its own chop value, and then recovered weight with optimal chop value is picked to plot the curve. As shown in the figure, the proposed scheme always outperforms the non-partitioned retransmission and achieves almost 97% of success of transmission for the given network and error parameter.

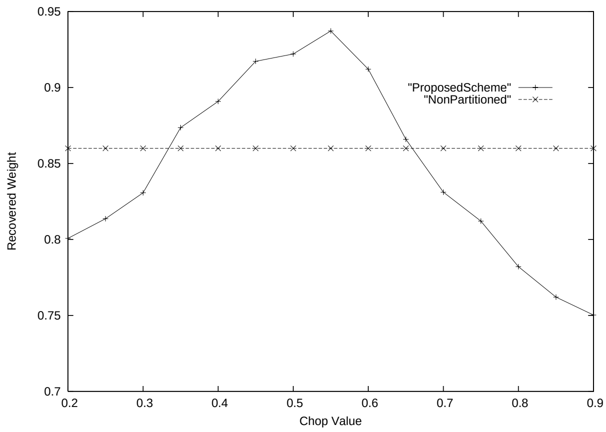

Figure 5. Recovered weight vs. chop value

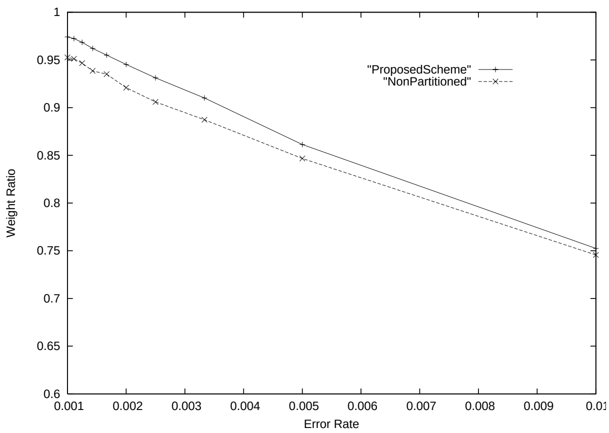

Figure 6. Total weight vs. error rate

V. Conclusion

In this paper, we have proposed and analyzed the performance of communication architecture capable of efficiently dealing with channel error on the wireless sensor network for the time-sensitive sensor application based on the IEEE 802.11 Wireless LAN standard. The proposed scheme makes AP always estimate channel status between itself and each sensor node, to avoid polling a node whose channel is not in normal condition. Once the packet transmission fails, it should be retried in a best-effort manner within its deadline. Finally, it can support the prioritized error recovery by dividing the DCF into two subperiods and differentiating their load. The experiment performed via simulation using SMPL shows that the proposed scheme, with this DCF partition, can improve the recovered weight compared with the non-partitioned scheme if a good chop value is found. For the given environment parameters, it shows about 8 % improvement when the chop value is 0.55. In addition, as for the sum of weights of successfully transmitted packets, the proposed scheme always outperforms non-partitioned scheme.

As a future work, we are to scrutinize the method to find the optimal chop value for the given importance distribution as well as other real-time communication parameters. Additionally, the work that combines the proposed communication architecture, will target to time-sensitive sensor application, with power management schemes.

ACKNOWLEDGMENT

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment) (IITA-2005-C1090-0502-0009).

REFERENCES

[1] Madden, S., Franklin, M., Hellerstein, J., Hong, W., "The design of an acquisitional query processor for sensor networks," *ACM SINGMOD*, 2003.

[2] Carley, T., Ba, M., Barua, R., Stewart, D., "Contention-free periodic message scheduler medium access control in wireless sensor/actuator networks," *Proc. IEEE Real-Time Systems Symposium*, Dec. 2003.

[3] Choi, S., Shin, K., "A unified wireless LAN architecture for real-time and non-real-time communication services," *IEEE/ACM Trans. on Networking*, pp.44-59, Feb. 2000.

[4] Adamou, M., Khanna, S., Lee, I., Shin, I., Zhou, S., "Fair real-time traffic scheduling over a wireless LAN," *Proc. IEEE Real-Time Systems Symposium*, pp.279-288, Dec. 2001.

[5] IEEE 802.11-1999, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 1999. also available at *http://standards.ieee.org/getieee802*.

[6] Caccamo, M., Zhang, L., Sha, L., Buttazzo, G., "An implicit prioritized access protocol for wireless sensor networks," *Proc. IEEE Real-Time Systems Symposium*, Dec. 2002.

[7] Vaidya, N., Bahl, P., Gupta, S., "Distributed fair scheduling in a wireless LAN," *6-th Annual Int'l Conference on Mobile Computing and Networking*, Aug. 2000.

[8] Sheu, S., Sheu, T., "A bandwidth allocation/sharing/ extension protocol for multimedia over IEEE 802.11 ad hoc wireless LANS," *IEEE Journal on Selected Areas in Communications*, Vol. 19, No. 10, pp.2065-2080, Oct. 2001.

[9] Shah, S. H., Chen, K., Nahrstedt, K., "Dynamic Bandwidth Management for Single-hop Ad Hoc Wireless Networks," *ACM/Kluwer Mobile Networks and Applications (MONET) Journal. 10*, pp.199-217, 2005.

[10] Liu, J., *Real-Time Systems*, Prentice Hall, 2000.

[11] Bai, H., Atiquzzaman, M., "Error modeling schemes for fading channels in wireless communications: A survey," *IEEE Communications Surveys*, Vol. 5, No. 2, pp.2-9, 2003.

[12] Bottigliengo, M., Casetti, C., Chiaserini, C., Meo, M., "Short term fairness for TCP flows in 802.11b WLANs," *Proc. IEEE INFOCOM*, 2004.

[13] Lee, J., Kang, M., Jin, Y., Kim, H., Kim, J., "An efficient bandwidth management scheme for a hard real-time fuzzy control system based on the wireless LAN," *accepted to LNCS: Embedded Systems for Ubiquitous Computing*, 2005.

[14] Gao, B., Garcia-Molina, H., "Scheduling soft real-time jobs over dual non-real-time servers," *IEEE Trans. Parallel and Distributed Systems*, pp.56-68, Jan. 1996.

[15] Fall, K., Varadhan, K., "Ns notes and documentation," *Technical Report, VINT project, UC-Berkeley and LBNL*, 1997.

[16] MacDougall, M., *Simulating Computer Systems: Techniques and Tools*, MIT Press, 1987.

Mikyung Kang received the B.S. and M.S. degrees at Dept. of computer science and statistics from Cheju National University, Jeju, Korea. Currently, she is a Ph.D. candidate at Cheju National University, Korea. Her research interests are real-time communication and wireless network.

She is a researcher of Key Technologies for telematics systems at ITRC (Information Technology Research Center) of Cheju National University, Korea.

Junghoon Lee received the B.S., M.S., and Ph.D. degrees at Dept. computer engineering, Seoul National University, Korea. His research interests include real-time communication and wireless network.

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 Dept. 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 Dept. of computer science, University of Texas at Austin.

Yongmoon Jin received the B.S. and M.S. degrees at Dept. of computer science and statistics, Cheju National University, Jeju, Korea. Currently, He is a Ph.D. candidate of Cheju National University, Korea. His major field of study is wireless network and telematics System.

He is an Associate Researcher of key technologies for telematics systems at ITRC of Cheju National University, Korea.

Gyung-Leen Park received B.S. in Dept. 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 an assistant professor at the University of Texas at Arlington. In 1998, he joined the Dept. 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 of Cheju National University, Korea.

Hanil Kim received the B.S, M.S, and Ph.D. degrees at Dept. of computer engineering, Seoul National University, Korea. His research interests include personalized service, agent system, and computer education.

In 1995, he joined the Dept. of computer education at Cheju National University, Korea where he is currently an associate professor. He is currently a visiting scholar at Colorado State University since 2005.