Active Caching: A Transmission Method to Guarantee Desired Communication Reliability in Wireless Sensor Networks

Dae-Young Kim and Jinsung Cho, *Member, IEEE*

***Abstract*—Due to the high packet loss rate during multi-hop transmissions in wireless sensor networks, more reliable end-to-end data transmission is desirable. Because wireless sensor network applications require various levels of communication reliability (CR), the end-to-end data transmission should satisfy the desired CR of the applications. In this letter, we propose a flexible loss recovery mechanism for sensor network applications with various CRs. The proposed scheme caches data packets at intermediate nodes over routing paths computed by CR to retransmit lost packets during multi-hop transmissions. Because the proposed scheme presents a tradeoff between end-to-end delays and memory requirements dependent on CR, it can be used flexibly in various sensor network applications.**

***Index Terms*—Loss recovery, reliable transmission, multi-hop transmission, wireless sensor networks.**

I. Introduction

RECENT advances in wireless communication have enabled multifunctional tiny nodes to construct a wireless network by themselves [1]. The network is called a wireless sensor network. The communication systems in end-to-end data transmission of wireless sensor networks employ a recovery mechanism for lost data during data transmissions because reliable data transmissions are required for various sensor network applications.

Two types of retransmission have been proposed for the recovery, namely end-to-end loss recovery (E2E) and hop-by-hop loss recovery (HBH). In these mechanisms, lost packets are retransmitted from a source node or an intermediate node. If a retransmit request for lost packets is sent to a source node, the end-to-end delay may increase because channel error accumulates exponentially over multi-hops [2]. The well-known HBH mechanisms are PSFQ [2] and RMST [3]. PSFQ is based on ACK message and RMST is on NACK message. In HBH, when intermediate nodes cache data packets into storage, retransmissions can be requested to an intermediate relay node to reduce end-to-end delays. Because sensor nodes have limited resources, however, it is difficult for all sensor nodes to find sufficient space in their routing paths to cache data packets. There is therefore a tradeoff between end-to-end delays and memory requirements.

Because data traffic on sensor networks requires a variety of levels of communication reliability (CR) depending on the application, a loss recovery method to guarantee the desired CR should be provided. Traditional loss recovery mechanisms consider only 100% reliability. In this letter, we propose a flexible loss recovery mechanism to guarantee various CRs and we discuss the tradeoff between end-to-end delays and memory requirements for various CRs. The proposed method can be widely used for the design of wireless sensor networks that require a variety of CRs.

Manuscript received January 5, 2009. The associate editor coordinating the review of this letter and approving it for publication was F. Theolegre.

The authors are with the Department of Computer Engineering, Kyung Hee University, Korea (e-mail: {kimdy, chojs}@khu.ac.kr).

Digital Object Identifier 10.1109/LCOMM.2009.090023

$RELIABLE-TRANSMIT(CR, i, p_i, P_{tx}(i-1), F(i-1))$

1. $P_{tx}[i] \leftarrow P_{tx}[i-1] \cdot (1-p_i)$
2. if $P_{tx}[i] > CR$
3. then $F[i] \leftarrow false$
4. else $F[i] \leftarrow true$
5. $P_{tx}[i] \leftarrow (1-p_i)$
6. cache data packets to a node n_i

Fig. 1. Active caching algorithm at i-th node, n_i.

II. The Proposed Scheme: Active Caching

As mentioned previously, E2E involves large end-to-end delays for 100% reliability because of high packet loss during multi-hop transmissions. To guarantee high reliability and minimal end-to-end delays, HBH caches data in every node over a routing path resulting in large memory requirements. When only some nodes cache data on a routing path, there exists a tradeoff between the end-to-end delays and the memory requirements. For applications which do not require 100% reliability, every node needs not cache data via HBH. When a target CR is given, we need a flexible method to guarantee the given CR while minimizing the memory requirement. In this section, we present such a method - active caching (AC).

The proposed scheme allows various CRs of application services. It determines positions where data caching occurs using a dynamic programming algorithm, which solves every subproblem just once and then saves its answer in a table to avoid the work of recomputing the answer [4]. If there are holes in sequence numbers of received data, a caching node recognizes packet loss [5]. The caching node sends a NACK message to a previous caching node along the path and the previous caching node retransmits lost packets selectively.

First, we define the problem and subproblems for the active caching as a dynamic programming algorithm to guarantee an end-to-end reliable data transmission as:

Problem: $P_{tx}(H) > CR$.
Subproblem: $P_{tx}(h) > CR$, where $h = 1, 2, \cdots, H$.

The packet delivery rate $P_{tx}(H)$ during total hop counts H should be greater than the desired communication reliability CR. To do that, the packet delivery rate $P_{tx}(h)$ during hop counts h in each hop should be greater than the CR. The key idea for solving the problem is to cache data packets if the probability of packet transmission does not satisfy the desired communication reliability. By solving the subproblems, we can solve the entire problem.

Fig. 1 shows the proposed active caching algorithm for loss recovery. Each node solves the subproblem using the tables for

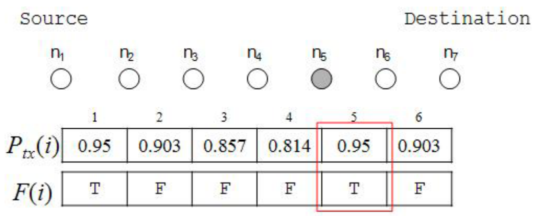

Fig. 2. An example of active caching.

the packet delivery rate $P_{tx}(i)$ until i-th hop and the caching flag of i-th node $F(i)$. Both $P_{tx}(i-1)$ and $F(i-1)$ of the tables are piggybacked in data packets and they are delivered to the next node. In a source node ($i=1$), $P_{tx}(1)$ is $1-p_1$ as the packet delivery rate at the 1st hop and $F(1)$ is $true$. *Line 1-3:* n_i calculates $P_{tx}(i)$ using $P_{tx}(i-1)$, where $P_{tx}(i)$ accumulates the packet delivery rate $1-p_i$ of i-th hop while packets are transmitted. After that, it compares $P_{tx}(i)$ with CR. If $P_{tx}(i)$ satisfies the desired CR, n_i is not a caching node ($F(i)$ is $false$). *Line 4-6:* If $P_{tx}(i)$ does not guarantee the desired CR, n_i becomes a caching node ($F(i)$ is $true$). In this case, $P_{tx}(i)$ compensates for its packet delivery rate as the reliability instead of accumulating $P_{tx}(i)$ and data packets are cached onto n_i's buffer. Each node runs the algorithm of Fig. 1 and the total active caching over a routing path is performed by the dynamic programming algorithm. Fig. 2 shows an example of the active caching when seven sensor nodes are deployed sequentially and they have an average 5% packet loss rate and 80% CR. Every node satisfies 80% CR and data caching occurs at n_5. When packet loss happens between a source node n_1 and the caching node n_5, the caching node requests retransmission to the source node. When packet loss happens between the caching node and a destination node n_7, the destination node requests retransmission to the caching node.

III. Analysis

A packet loss rate occurs due to wireless link and contention errors. Since all the packets are destined to the sink node in wireless sensor networks, the contention error in links close to the sink node may increase. To model the packet loss rate at i-th hop, we assume the uniform link error p and the contention error which is proportional to the square of transmission hop counts.

$$p_i = p + \alpha i^2, \tag{1}$$

where α is the contention failure factor. Then the packet delivery rate during h hops from the s-th node is

$$P_{tx}(s,h) = \prod_{i=s}^{s+h-1} (1-p_i). \tag{2}$$

Data caching occurs when $P_{tx}(s,h)$ is lower than CR. When the number of nodes N over a route and CR are given, the hop counts h from a caching node s and the number of caching nodes N_c are obtained by the function in Fig. 3. Φ represents a set of (s,h) tuples and the (s,h) tuples are used to compute the retransmission counts of lost packets. For example in Fig. 2, $\Phi = \{(1,4),(5,2)\}$.

$$\Phi = \{(s_j,h_j) \mid j = 1,\cdots,N_C\}. \tag{3}$$

```
CalcHopCounts(N, CR)
n ← 1, s ← 1, h ← 1, N_c ← 0
Φ = φ
loop: n < N
    if P_tx(s,h) > CR
        then n ← n+1, h ← h+1          //no caching
        else h ← h−1                   //caching
            if (h = 0)
                then h ← 1, n ← n+1
            add (s,h) to Φ, N_c ← N_c + 1
            s ← n, h ← 1
end loop
if (h > 1)
    then add (s, h−1) to Φ, N_c ← N_c + 1
```

Fig. 3. Function to obtain (s,h) tuples.

If the retransmission counts for h hops from a caching node s is given by $\psi(s,h)$, the total retransmission counts $E[C]$ between a source node and a sink node are represented by the sum of $\psi(s,h)$ as

$$E[C] = \sum_{j=1}^{N_c} \psi(s_j,h_j). \tag{4}$$

Because the retransmitted packets can also experience transmission failure, we should consider repeated retransmissions for $\psi(s,h)$. Let $\Gamma_f(j,s,h)$ indicate the number of transmitted packets at the j-th retransmission. Then $\psi(s,h)$ can be represented as

$$\psi(s,h) = \sum_{j=1}^{\infty} \big(h \cdot \Gamma_f(j,s,h) \cdot P_{tx}(s,h)\big). \tag{5}$$

If we let $\Gamma_s(k,s,h)$ be the number of successfully transmitted packets among k packets during h hops from node s, $\Gamma_f(j,s,h)$ can be represented recursively as

$$\Gamma_f(j,s,h) = \Gamma_f(j-1,s,h) - \big[\Gamma_s\big(\Gamma_f(j-1,s,h),s,h\big)\big]^1, \tag{6}$$

where $\Gamma_f(0,s,h) = K$ and K is the number of total packets which is generated in a source node.

The number of successfully transmitted packets $\Gamma_s(k,s,h)$ can be calculated by the probability of successful transmission of Bernoulli trials $P_s(k,m,s,h)$ as

$$\Gamma_s(k,s,h) = \sum_{m=1}^{k} m \cdot P_s(k,m,s,h). \tag{7}$$

If m data packets are transmitted successfully among k packets to deliver across h hops from a caching node s, the probability of successful transmissions can be obtained by Bernoulli trials as

$$P_s(k,m,s,h) = \binom{k}{m} \cdot P_{tx}(s,h)^m \cdot \big(1-P_{tx}(s,h)\big)^{k-m}. \tag{8}$$

The memory requirement B is defined as the caching rates of intermediate nodes including a source node. It is computed by N_c and the number of relay nodes over a routing path:

$$E[B] = \frac{N_c}{N-1}. \tag{9}$$

A high $E[C]$ indicates large end-to-end transmission delays and $E[B]$ represents the memory requirements of buffers on the data transmission routes. Because both $E[C]$ and $E[B]$ can be estimated by CR of traffic through Eq.(4) and Eq.(9), a flexible data transmission system can be designed.

[1] $[x]$ is n, in case of $n-0.5 \le x < n+0.5$

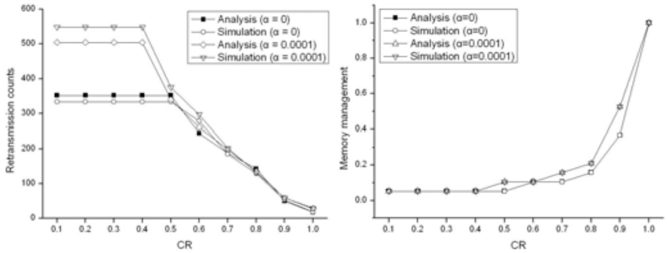

Fig. 4. Validation of our analysis (p=0.03).

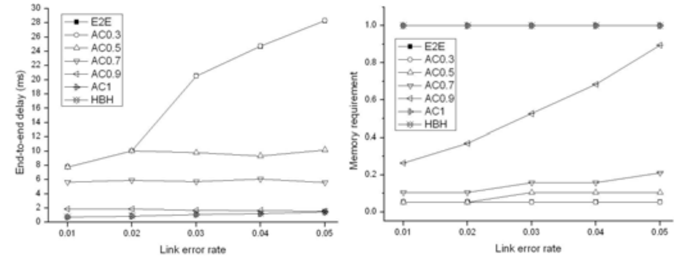

Fig. 5. Performance comparison of E2E, HBH, and AC.

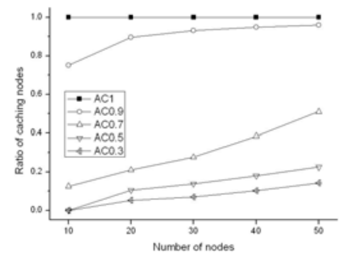

Fig. 6. The ratio of caching nodes.

IV. Evaluation

In this section, we validate the analysis through simulations and compare the performance of active caching (AC) with that of E2E and HBH. For the simulation, we assume 20 sensor nodes are deployed sequentially and the wireless channel has both link and contention error as described in Section III. The contention failure factor α is determined as 0.0001 by considering total hop counts. So, p_i in Eq.(1) ranges from 0.03 to 0.07 when p is 0.03 in our experiments. The sensor nodes employ AODV as a routing protocol. Assuming a packet is 30 bytes and the data rate is 250kbps, we perform the analysis and simulation by varying CR from 10% to 100%. AC with CR from 0.1 to 1 is expressed as AC0.1 to AC1.

Fig. 4 shows the results of the analysis and the simulation of the retransmission counts and the memory requirements when a source transmits 40 packets. The results of the analysis and the simulation show an average of 94% similarity. Fig. 4 also represents the tradeoff as mentioned earlier. The high CR requires a high memory requirement for reliability and it decreases the retransmission counts. When the memory requirement is the lowest, the retransmission counts are the highest and AC runs as E2E. In short, we can design wireless sensor networks that take the desired CR and memory requirements into consideration through the proposed active caching.

Fig. 5 shows the performance comparison of E2E, HBH, and AC. Because AC with the highest memory requirement caches data to every intermediate node, it operates as HBH. When AC does not perform data caching, it operates as E2E. That is, AC switches between HBH and E2E while showing the performance tradeoff between them. In addition, it has a tolerable end-to-end delay to minimize the memory requirement depending on CR. In Fig. 5, the end-to-end delays of E2E increase when the wireless channel has a high link error rate. However, the end-to-end delay of AC maintains similar values because AC increases the memory requirements to ensure CR. An evaluation has been performed for 10 and 50 nodes deployed over a route, and the results are similar to the case of 20 nodes. These results have been omitted due to the page limitation.

Fig. 6 shows the ratio of caching nodes over relay nodes. Because the contention error increases when the density of nodes increases, the ratio of caching nodes increases when the number of sensor nodes increases.

V. Conclusion

Wireless sensor networks transmit data through multiple hops. End-to-end data transmission must recover lost data for reliable data transmissions. Active caching (AC) provides more flexible end-to-end delays and memory requirements for a given reliability than the existing recovery mechanisms (i.e., E2E, HBH). By using the proposed dynamic loss recovery with active caching, a flexible end-to-end data transmission system can be designed.

Acknowledgment

This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MEST) (No. R01-2008-000-20029-0) and by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Advancement) (IITA-2009-(C1090-0902-0002)).

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102-114, Aug. 2002.

[2] C. Y. Wan, A. T. Campbell, and L. Krishnamurthy, "PSFQ: a reliable transport protocol for wireless sensor networks," in *Proc. ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 1-11, Sept. 2002.

[3] F. Stann and J. Heidemann, "RMST: reliable data transport in sensor networks," in *Proc. IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 102-112, May 2003.

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, vol. 1, pp. 323-369. The MIT Press, 2001.

[5] H. Karl, *Protocols and Architectures for Wireless Sensor Networks*, pp. 359-411. John Wiley & Sons, 2005.