Cushion: Autonomically Adaptive Data Fusion in Wireless Sensor Networks*

Jungmin So Jintae Kim Indranil Gupta
University of Illinois at Urbana-Champaign
{jso1, kim28, indy}@cs.uiuc.edu

1. Introduction

Data fusion is an important service for sensor network applications. The goal of data fusion (or aggregation) is to aggregate data from independent sensors to compute useful information, such as the average of all the sensor readings, the maximum value among the sensor readings, or the number of sensors that detect an event. In data aggregation, reliability is often measured by the participation ratio, i.e., the number of participating nodes over the total number of nodes.

The data aggregation challenge addressed in this paper is: how does one achieve a desired level of reliability at all times, with low message overhead, in spite of time-varying unreliability in the underlying networks? In other words, can we design an *autonomically* and *automatically* adaptive protocol that maintains a high reliability, even as nodes fail and recover, and as the lossiness of links varies over time, without the need to configure parameters online or the need for human intervention? In addition, energy conservation requires the adaptive protocol to have low control overhead and actual aggregation message overhead.

In a typical in-network aggregation problem, data originates from multiple *source nodes*, and moves towards single *sink node* or root node. Along the way, i.e., *inside* the network, the data may be partially aggregated, thus reducing message overhead. Two well-known classes of solutions to this problem are: tree-based aggregation and multi-path aggregation. Our autonomically adaptive data fusion solution, called *Cushion*, contains two protocols that span a continuous spectrum between these two design points and can further increase reliability over multi-path approach using redundant transmissions. We motivate the design of Cushion by first discussing the complementary advantages and disadvantages of the two design points.

Tree-Based Aggregation: In tree-based approaches, e.g., [2], a tree is constructed and maintained, with the sink node as the root. Data converges from the source nodes towards the sink along the edges of the tree. However, unreliable channels, energy depletion, and node failures, can result in momentary loss of entire subtrees of readings, while the tree is reorganized. Failures closer to the root affect the reliability very drastically. Finally, if the rate of failures is comparable to the rate of tree reorganization, a *thrashing-like* behavior could produce a consistently low reliability all the time.

Multi-Path Aggregation: Multi-path data fusion protocols take advantage of broadcast communication in the sensor networks [3]. Reliability can be improved by broadcast communication because each node can send data to multiple parent nodes without incurring additional overhead. While this approach inevitably accompanies the problem of detecting duplicate messages from a node, Flajolet and Martin's counting algorithm [1] provides the framework to solve this duplicate problem. When all nodes in the network (except the sink) are sources, then the overhead of multi-path aggregation is same as tree-based aggregation (each node transmits once). However, if only a subset of nodes are sources, then multi-path aggregation may have larger overhead compared to tree-based aggregation, since nodes that are not part of the tree may participate in forwarding packets.

New Approach in Cushion: In this paper, we present two new aggregation protocols that adopt a *probabilistic approach* to realize autonomic and automatic adaptivity. Initially, a spanning tree rooted at the sink node is established in the network. All nodes except the sink are classified into two types: nodes along the spanning tree whose leaf is a source, or nodes outside the spanning tree. The most important parameter, for achieving adaptivity, is the *redundancy level* $p(> 0)$. When $p = 0$, the Cushion protocols operate as a tree-based protocol. In this case, node that are outside the spanning tree do not forward the packets from the source. (Nodes that are on the spanning tree forward packets from the source, regardless of p.) If the sink node does not obtain the desired reliability due to failures in the network, the sink attempts to increase the redundancy levels of the sources in the system. On the other hand, if reliability rises above the target reliability, e.g., due to node recoveries, the sink reduces the redundancy level.

[1]This work was supported in part by NSF CAREER grant CNS-0448246 and in part by NSF ITR grant CMS-0427089.

0-7803-9466-6/05/$20.00 ©2005 IEEE MASS 2005

As a result of this reduction, the system might converge to a tree-like configuration, when the channel condition allows the sink to achieve the target reliability. As a result of this, the emergent behavior throughout the system is a minimization of the message overhead while maintaining a certain level of reliability, even as nodes fail and links become lossy.

If the redundancy level at each source in the system is increased to $p = 1$, the Cushion protocols reduce to a multipath protocol, where multiple parents forward packets from the source to increase reliability. When $0 < p < 1$, nodes outside of the spanning tree forwards a packet with probability p, and thus message overhead is adjusted between the tree-based and multi-path approach in order to satisfy the required reliability at a necessary cost. If the desired level of reliability is not met even when $p = 1$, then p can further be increased over 1. When $p > 1$, it translates into *redundant transmission*. For example, if $p = 3$, then each intermediate node (including nodes on the spanning tree) transmits a packet three times, to further increase reliability. The adaptive behavior of Cushion is illustrated in Figure 1.

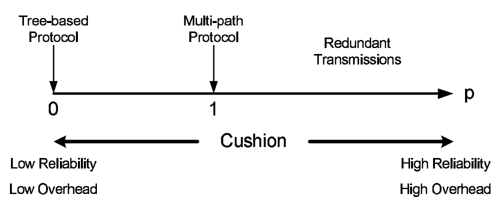

Figure 1. Illustration of adaptive behavior of Cushion. The Cushion protocols achieve desired reliability level by controlling message overhead using the redundancy level p.

Depending on whether the sink node adjusts the redundancy level on a system-wide basis (i.e., at all sources) or on a per-source basis, we obtain two flavors of adaptive aggregation protocols. Due to the lack of space, we describe our protocols very briefly and present a part of our simulation results. For further details, refer to [4].

2. Cushion: Adaptive Data Fusion Protocols

The goal of the Cushion protocols is to maintain a desired level of reliability (participation ratio), regardless of dynamically changing network conditions, by controlling the amount of message overhead. Moreover, the amount of message overhead should not be more than needed for achieving the desired reliability level.

In order to achieve this goal, the Cushion protocols move between tree-based and multi-path protocols based on channel condition. For the basic components of an aggregation protocol, the Cushion protocols share similarities with other aggregation protocols. For example, the tree establishment, routing and aggregation strategies are similar to [2]. Multi-path routing and duplicate detection schemes are similar to [3]. Here we omit the details due to lack of space, and describe how Cushion protocols adapt themselves to changing network conditions to achieve desired reliability. Details of the basic components are described in [4].

The main idea of adaptive aggregation is to have the sink node measure the reliability level, and control the message overhead by sending out CONTROL messages to the network. In particular, the sink node controls the *redundancy level* $p(> 0)$ of the sources in order to control the message overhead.

According to the redundancy level p, the protocols behave as the following.

- $p = 0$: The protocols behave similar to a tree-based protocol. Only the nodes in the spanning tree forward the packets, in its corresponding round that matches its depth. (A *round* is defined in the previous section.) Other nodes do not forward packets.

- $0 < p < 1$: Nodes that are not on the spanning tree forwards the packet with the probability p in its corresponding round. The nodes in the spanning tree still forward with 100% probability.

- $p = 1$: All nodes that receive packets in the previous round forward the packet in the corresponding round. The behavior of the protocols becomes similar to [3].

- $p > 1$: Nodes forward packets more than once. That is, each node transmits the packet with 100% probability in $\lfloor p \rfloor$ rounds, and transmits with probability $p - \lfloor p \rfloor$ in the following round.

The reliability and overhead are both increased when the redundancy level increases, and drops down as the redundancy level is decreased. So redundancy level is the key parameter in controlling overhead to maintain desired reliability level in Cushion. Depending on how the redundancy level is assigned to sources, there are two flavors of Cushion protocols: Simple adaptive protocol and Node-aware adaptive protocol.

In the Simple Adaptive protocol, a single redundancy level p is assigned for all nodes, and p is controlled by the sink node. The sink node monitors the percentage of packets it receives, and decides whether to increase or decrease p. If the sink node does not obtain the target reliability, it increases p. On the other hand, if the sink node obtains the target reliability, it starts decreasing p. Whenever the sink node decides to change the redundancy level, it selects new p and broadcasts the new redundancy level in a CONTROL

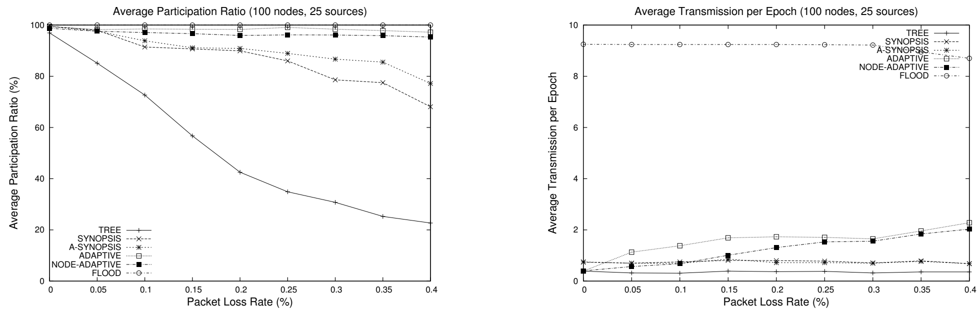

Figure 2. Average participation ratio (reliability) and average transmissions per epoch (overhead) of protocols for the scenario with 100 nodes and 25 sources.

message. If the channel condition is very good, the adaptive protocol will converge to a tree-based protocol. As the channel condition worsens, the protocol will increase redundancy to maintain the target reliability. To achieve the target reliability quickly when the reliability level is lower than the target, we use a Multiplicative Increase Additive Decrease (MIAD) approach, where p is increased multiplicatively and decreased additively.

In Node-Aware Adaptive protocol, a separate redundancy level p is assigned for each node, instead of having a common p. The sink node maintains the redundancy level for each source, and also the *history* of aggregation for the last w epochs. The history records whether a source has participated in the data aggregation or not, in each epoch. Using the history, the sink node can find out the participation rate of each source in the recent past. When the target reliability is not obtained, only m nodes with the worst participation rate are assigned an increased redundancy level. This protocol can be more efficient than the Simple Adaptive protocol, especially in situations where channel condition varies in different regions.

3. Experimental Results

We have used the ns-2 simulator to evaluate the performance of the proposed Cushion protocols. The sensor nodes have transmission range of 10m, and 100 nodes are randomly placed in a 50m $\times$ 50m area. Among them, 25 nodes are selected as sources. A sink node is placed in the center, which broadcasts a QUERY message every 10 seconds to gather information from the source nodes. The source nodes reply to the QUERY message by sending ANSWER messages in every second. The type of aggregate we use in our simulations is COUNT, which calculates the number of sources. Since COUNT is a duplicate-sensitive aggregate, our protocols can be applied to other types of aggregate as well.

Under this simulation setup, we have evaluated our proposed protocols, Simple Adaptive (denoted as ADAPTIVE) and Node-Aware Adaptive (N-ADAPTIVE). For comparison, we have also simulated the tree protocol (TREE) in [2], the two synopsis protocols (SYNOPSIS and A-SYNOPSIS) in [3], and flooding protocol (FLOOD) which is also described in [3].

Figure 2(a) shows the average participation ratio, which is a measure of reliability. FLOOD, ADAPTIVE, and N-ADAPTIVE maintain participation ratio over 90%, even when the packet loss rate increases up to 40%. The reliability of the TREE protocol drops down at the fastest rate, followed by SYNOPSIS and A-SYNOPSIS. Figure 2(b) shows the overhead of each protocols. In the figure, TREE, SYNOPSIS, A-SYNOPSIS and FLOOD show no change in the overhead. However, for ADAPTIVE and N-ADAPTIVE protocols, the overhead is increased as the packet loss ratio increases. From the two figures, we can see that the Cushion protocols maintain desired reliability regardless of the channel condition, by controlling the amount of overhead. More simulation results, including results that differentiate ADAPTIVE and N-ADAPTIVE protocols, are shown in [4].

References

[1] P. Flajolet and G. N. Martin. Probabilistic counting algorithms for database applications. *Journal of Computer and System Sciences*, 1985.

[2] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. Tag: A tiny aggregation service for ad hoc sensor networks. In *OSDI*, 2002.

[3] S. Nath, P. B. Gibbons, S. Seshan, and Z. R. Anderson. Synopsis diffusion for robust aggregation in sensor networks. In *ACM SenSys*, 2004.

[4] J. So, J. Kim, and I. Gupta. Cushion: Autonomically adaptive data fusion in wireless sensor networks. Technical report.