Simba: Similar-evolution Based Aggregation in Wireless Sensor Networks

Jin CUI, Fabrice VALOIS
Université de Lyon, INSA-Lyon
INRIA, CITI, F-69621 Villeurbanne, France
Email: {jin.cui,fabrice.valois}@insa-lyon.fr

Abstract—Data aggregation is an important mechanism to reduce energy consumption in Wireless Sensor Networks (WSNs). By investigating spatial and/or temporal correlation of raw data, sensor nodes can aggregate raw data to a meaningful digest instead of directly sending raw data to sink, this process is considered as data aggregation. Several aggregation works focus on the raw data, they use raw data to cluster the nodes or to do aggregation. While analysis of datasets of real projects shows that some nodes perform similar evolution. Thus we propose a raw data-independent aggregation, i.e., Similar-evolution Based data Aggregation (Simba), to consider the evolution of data rather than the raw data. Simba creates a group out of isolated nodes, nodes in the group can cooperatively execute data aggregation, this process reduces the energy consumption on each node. Besides, similar evolution of nodes guarantees the recover accuracy. Our experiments demonstrate that Simba can save more than 91% energy comparing no aggregation, and save more 30% energy than original aggregation functions, and Simba can recover data with higher fidelity comparing with the works relying on raw data.

Keywords—Data aggregation; temporal-spatial correlation; similar evolution; wireless sensor networks.

Wireless Sensor Networks (WSNs) have attracted lots of attentions from both academia world and industry for the past ten years. Due to the properties of flexibility, self-organization and so on, a variety of applications are developed based on WSNs. However, energy constrain and network capacity constrain limits the performances of such networks. Data aggregation [1], which is a way to save energy and network capacity, has been investigated frequently. It is a data gathering strategy with the idea of using data correlation to reduce the amount of data, thereby saving energy and reducing capacity consumption.

As a way to investigate the data correlation, raw data has been considered frequently. A lot of aggregation schemes are proposed to organize sensor nodes by the raw data, such as characteristic aggregation [2]. The authors highlight that similar raw data can be used to organize nodes, i.e., if nodes have similar raw data (defined by threshold), they can be grouped together and achieve data aggregation. However, by investigating real datasets, we analyse two properties: 1) abnormal data often occurs in set of raw data; 2) nodes show similar evolutions. The existing of abnormal data will impact the accuracy of works relying on raw data. While similar evolution is more stable than raw data, thus it has the potential to guarantee the accuracy.

Therefore, we propose a data-independent aggregation scheme: Similar-evolution Based Aggregation (Simba). Simba takes benefits from groups, and the groups are built by similarity of evolution. In a network, Simba introduces two phases: set-up phase and aggregation phase. Set-up phase is group forming phase, all the nodes use a vector (\(\vec{Rc}\)) to characterize evolution. By communicating with neighbours, nodes holding similar evolution form a group. A group leader will be selected to represent the group. Aggregation phase is aggregation function executing phase, when data of group leader are recovered by sink, the data of other nodes in the same group can be easily computed. We provide aggregation functions, A-ARMA [3], polynomial aggregation [4] and average, to test the performance with and without Simba. The simulation results from Matlab and WSNet [5] show that, Simba can save more 30% energy than original functions, and save more than 91% energy comparing with no aggregation. Meanwhile, sink can recover data with high fidelity according to RMS computation. Comparing with characteristic aggregation [2], Simba can recover the data more accurately.

Our main contributions are in twofold:

- We investigate the real datasets, and analyse two key properties: 1) abnormal data often occurs in raw data; 2) several nodes show similar evolution.
- We propose Simba, which groups nodes by the similarity of evolution. In a group, nodes execute aggregation functions in turn. The experiments show that Simba reduces energy consumption and guarantees the recover accuracy in the meanwhile.

I. CAPTURING EVOLUTION

Linear regression investigation [6] indicates that straight lines can be used to approximate time series, and the straight lines can be seen as evolution in our case. The raw data of a sensor node can be seen as a set of \((y, i)\), in which \(y\) is the data value and \(i\) is the corresponding time sequence. Thus, linear regression can be simply expressed as: \(y = Rc \cdot i + \beta\), where \(Rc\) is the slope of the straight line, and \(\beta\) is the intercept. In our case, we consider \(Rc\) to demonstrate the evolution.

In order to precisely capture the evolution, we use piecewise linear regression [7]. That is to say the linear regression

---

1Evolution here means the data trend. Similar-evolution is the similar-trend in this paper.
will be computed in segments. Regarding a segment, regression coefficient $R_c$ is computed by:

$$ R_{c_{seg}} = \frac{\sum_{i=1}^{S_i} (i - \bar{i})(y_i - \bar{y})}{\sum_{i=1}^{S_i} (i - \bar{i})^2} \quad (1) $$

where $S_i$ is the segment length (number of data in a segment), $\bar{i} = \frac{1}{S_i} \sum_{i=1}^{S_i} i$, $\bar{y} = \frac{1}{S_i} \sum_{i=1}^{S_i} y_i$, and $S_{eg}$ denotes the current segment. With more segments, node will generate a coefficient vector $\bar{R}_c = [\cdots, R_{c_{seg} - 1}, R_{c_{seg}}]$, which can characterize the data evolution more precisely.

II. MEASURING SIMILARITY

Cosine similarity [8] is a usual measurement, which shows the cosine of the angle between two vectors. The cosine similarity between $\vec{R}_c^i$ and $\vec{R}_c^j$ can be formulated as follow:

$$ C_s(\vec{R}_c^i, \vec{R}_c^j) = \frac{\vec{R}_c^i \cdot \vec{R}_c^j}{\sqrt{(\vec{R}_c^i \cdot \vec{R}_c^i)}(\vec{R}_c^j \cdot \vec{R}_c^j)} \quad (2) $$

where $\vec{R}_c^T$ denotes the transpose vector of $\vec{R}_c$. The cosine similarity values range between $[0, 1]$, where a value of $0$ means that the vectors are dissimilar and a cosine similarity value close to $1$ means that the vectors are similar. Giving a threshold $t_{thcs}$, cosine similarity can provide us an explicit index to show whether two vectors are similar or not. For clarity, we provide the following definition:

**Definition 1:** We define the similarity between two vectors ($\vec{R}_c$) as:

$$ sim(\vec{R}_c^i, \vec{R}_c^j) = \begin{cases} 1 & \text{if } C_s(\vec{R}_c^i, \vec{R}_c^j) \geq t_{thcs}, \\ 0 & \text{otherwise}. \end{cases} \quad (3) $$

Since vector $\vec{R}_c$ represents the data evaluation, we consider that two data evolutions are similar if and only if their $sim(\cdot, \cdot)$ value is equal to $1$. Note that $t_{thcs}$ should be bigger than $0.5$ to avoid two orthogonal vectors being considered similar.

**Theorem 1:** The similarity has transitive property. If $\vec{R}_c^i$ is similar to $\vec{R}_c^j$ and $\vec{R}_c^k$, thus $\vec{R}_c^j$ is also similar to $\vec{R}_c^k$, i.e.,

$$ \forall i, \ sim(\vec{R}_c^i, \vec{R}_c^j) = 1 \& sim(\vec{R}_c^i, \vec{R}_c^k) = 1 \Rightarrow sim(\vec{R}_c^j, \vec{R}_c^k) = 1, \forall j, k \quad (4) $$

To the page limitation, we omit the proof process.

III. SIMILAR-EVOLUTION BASED AGGREGATION

There are two phases of Simba: set-up phase and aggregation phase. Set-up phase focuses on organizing nodes: nodes having similar evolutions are grouped together. Aggregation phase focuses on executing data aggregation, where a Group Leader (GL) represents the group and is able to transmit the aggregated packet.

During the set-up phase, sensor nodes aim to group together. First, each node computes its coefficient vector $\bar{R}_c$ using Eq. 1, and broadcasts it to its neighbors. When the neighbors receive $\bar{R}_c$, they will calculate the similarity using Eq. 3, and the information are stored in a neighbor table. When node verifies the similar neighbors (value of Eq. 3 equals $1$), it will send all the similar neighbors to sink.

In aggregation phase, nodes in one group will share the aggregated information (this information is provided by the associated aggregation function). Sink will define the leader sequence, and send all the information to the first leader. Assuming nodes A, B and D belong to the same group, and node D is the current leader which aggregates information as the original aggregation process discussed above. When node D checks that the accuracy is beyond $t_{h_{err}}$, it sends new aggregated data to sink and the next leader (assuming node A is the next leader). Then node A will use the received aggregated packet to predict data and compare with new raw data, and then repeats the process as node D.

Due to the similar evolution, nodes in one group share the aggregated information, and the group is represented by one node (i.e. GL). Meanwhile, each node in the same group becomes leader in turn, thus each node will check the aggregated packet as its raw data at each step, thereby guaranteeing the accuracy.

IV. PERFORMANCE EVALUATION

Related works always pay attention to raw data, but few aggregation methods, to the best of our knowledge, focus on data evolution. Characteristic correlation [2] propose to cluster the nodes that own similar raw data, and we thus select it as the state-of-the-art benchmark for Simba. The nodes holding similar raw data form a virtual cluster in [2], and the average value is representative for the cluster. When sink receives the representative value, it retrieves the value as all the cluster members’ data.

To compare with Simba, we simulate characteristic correlation in Matlab using dataset $T_h$. The categorizing range is actually the error threshold in Simba, referred from the literature [2]: we use $0.5^\circ C$ and $1^\circ C$ to test the recovered accuracy. In addition, as proposed in [2], the cluster needs to train several data in order to keep it more stable, thus the first $100$ data are used for this purpose. Average is used as aggregation function for Simba and characteristic correlation. We calculate accuracy (RMS error between recovered data and raw data) for every $10$ data, to show the performance of two aggregation methods.

Fig. 1 demonstrates the RMS error per $10$ data for characteristic correlation and Simba: we can see Simba leads to a lower error than characteristic correlation. Moreover, with different categorizing range ($0.5^\circ C$ and $1^\circ C$), characteristic correlation shows a quite unstable RMS error. This is because characteristic correlation clusters nodes owning similar raw data, and when the raw data changes abruptly, the accuracy will be effected directly. For Simba, it uses evolution to group nodes together, the evolution is more stable than raw data. Besides, in characteristic correlation, sink retrieves the average values as all cluster members’ data, it does not consider the
Figure 1. Comparison analysis between characteristic correlation and Simba, considering error threshold as $0.5^\circ C$ and $1^\circ C$.

difference between nodes. While for Simba, sink uses the mean difference to retrieve data, which can guarantee the accuracy further.

REFERENCES


