The algorithm for distribution of large-size data in the Wireless Ad-Hoc Sensor Network

Marcin Golański, Radosław Schoeneich and Michał Siwko

Institute of Telecommunications, Warsaw University of Technology
ul. Nowowiejska 15/19, 00-665 Warszawa, Poland

Abstract: The article describes algorithm for distribution of large-size data in the Wireless Ad-hoc Sensor Network (WASN). The algorithm provides replication and large-size data allocation using the concept of decomposition into small pieces and random distribution in the WASN. The basic assumption is the large size of stored data, which means that the data size is much larger than the capacity of a single node. Thus, the data should be divided into small parts, with the result that can be deployed on multiple nodes. The authors also established a dynamic network partitioning, which implies data replication and partition-tolerant data storage. Proposed solution uses tree-based structure of the connections between nodes. This allows to exchange information and distribute parts of data. The article presents an algorithm, and performance measurements made using low level network metrics, such as total number of transmitted packets, and high level data availability ratio.

Keywords: large data distribution, Wireless Ad-hoc Sensor Networks

1. Introduction

Nowadays Wireless Ad-hoc Sensor Networks are dynamically explored. The WASN is a collaboration of tiny nodes. Nodes are connected via wireless links in the spontaneous manner. The WASN characterizes not only a changeable structure of the network, but also by attributes of the sensor node like: bandwidth, memory storage capacity and computational performance. Severe limitations of any single node is a reason for nodes grouping for the collaboration. Although the fast technological progress causes performance increasing and size minimization of each node, the
collaboration is still much more effective in terms of resources capacity enhancement. It can be understand as the overall summarization of the nodes resource.

Those mechanisms, which are relatively simple and well known to computer networks, cannot be easily implemented in an environment of small sensors.

Due to small sizes of the node and spontaneous structure of the network, the WASN may be used in many possible applications. It can be applied in rescue and emergency situations, where traditional communication infrastructure is unserviceable eg. in earthquake which breaks all wired connections, in small-populated and hardly-accessible terrains, or in the modern warfare theatre. The small size of the node makes possible to wear many of them on rescuers or solders uniforms, helmets and associated equipments. The common carrier makes possible creation of quasi-static collaboration of small nodes with huge overall performance. This feature enables to exchange and storage between actors large-size data such as: recordings, photographs, even movies or shared files. This significantly increases the potential for rescue team or military troops.

The major contribution of this paper is the algorithm for allocation and replication of large-size data in quasi-static wireless ad-hoc sensor network using tree-based distribution structure. As mentioned earlier, the basic assumptions of the presented publication are: (a) inability to store the large-size data for a single node, and (b) the possibility of network partitioning.

The outline of this paper is as follows: Section 2 reports on the main assumptions. Section 3 presents in detail the algorithm. Next two sections describes assumption, metrics and simulation results. The paper is concluded in Section 6.

2. Main assumptions

The main objective of the protocol described below is the distribution of data, for example in the form o file, throughout the Wireless Ad-hoc Sensor Network. Data size was assumed as \( L \) and network consist of \( n \) nodes. The capacity \( C_n \) of each node is determined in advance. Therefore, main assumptions for the proper algorithm execution are:

- We can organize the data \( L \) into \( m \) pieces, each of them has defined and constant capacity. For the purposes of algorithm these pieces are called quantum. The relationship between \( L \) and \( m \) is as follows:
\[ L = m \cdot q \]  \hspace{1cm} (1)

where \( q \) is the symbol of a single quantum. Capacity of node \( n \) is:

\[ C_n = k \cdot q \]  \hspace{1cm} (2)

where \( k \) is the number of quantum that can be placed in selected node.

- Each node maintain current list of neighbours. Therefore always we can send the data in reliable manner using single-hop approach.
- The algorithm operates in a short, finite time \( T \), in which the network topology, understood as connections between nodes, does not change.
- We define connections between nodes as edges of a consistent graph. Nodes respectively represent graph vertices.

Each node has a special memory in addition to his capacity \( C_n \). It allows to put at least a single quantum and routing information in a temporary buffer.

3. Data Distribution Algorithm

The Protocol operates in two stages: (a) injecting data into the network and (b) evaporate data from the network. Each of them is a single and rapid implementation of data transfer mechanisms. At the end of its operation, any routing information, and buffers are removed from nodes memory.

3.1 Data injection phase

This phase begins in root node with initialization the algorithm. First, a unique network-wide identifier \( IDa \) for the injectable data is chosen randomly. Then, in the first step, we build a connection tree from the root node to the end nodes, called leaves, on which data will be placed as shown on Figure 1.

It is carried out by the flooding the network of a special control packet. Subsequent nodes forward the packet to its neighbours. Thus they form a simple routing table in
memory of each node. It contains information about the neighbor who sent the packet - IN direction, and which have been sent further - OUT direction. Each leaf checks whether has sufficient capacity Cn to accept at least one quantum of data. If not, then sends the information back, in form of special reject packet, in the IN direction. It removes leaf from the parent node routing table by delete the selected entry. In this way, we create a map of connections used in the next step of the algorithm – data transmission.

![Diagram of the connection tree in data injection phase](image)

**Figure 1. Connection tree in data injection phase**

### 3.2 Data transmission and replication

The data, in form of quanta quantum, flowing down the tree from the root to the leafs according to the routing tables. Successive quanta are labelled sequentially by identifiers IDb. We assume a data redundancy to protect against data loss, for example as a result of a node separation from the network. Thus, each quantum is duplicated. Original copy is recognized by the odd number, while duplicated copy by the even respectively. Therefore, the full identification of each quantum consists of two parts: (a) the data identifier and (b) the quantum identifier – IDa.IDb.

The particular node receives from the IN direction the quantum. Then according to the routing table one of the OUT directions is chosen randomly. This mechanism is repeated for both – the original and duplicate. It provides the best statistical data distribution in the network. Moreover it decreases probability of data loss.
3.3 Data evaporation phase

This phase include mechanisms similar to those described in the preceding chapter. In addition to the reverse direction of data transmission, the main difference is the simplified data transfer to the root. It begins immediately after receiving of the control packet by node. It is important that root node can be chosen each time depending on the needs.

Described protocol does not maintain any information on the nodes after the operation. Between described phases, injection and evaporation, nodes resources can be used for other purposes. Thus, it is possible to reconfigure the network between algorithm phases. Injected data should preserve the integrity regardless of the actual network state.

4. Simulation and methodology

Our studies focused on the data distribution parts performance. We implemented the Algorithm using C# language for WSN Simulator [3] by Sourendra Sinha. Moreover for the simulation purpose a dedicated engine, which is activated in every node has been used.

The purpose of the simulation was to investigate the performance of the data injection phase using high and low level metrics which are useful in our algorithms examination.

In simulation scenarios nodes were randomly deployed in the 500m x 500m field. The range of each node was 125m assuming full battery capacity. The paper presents results of our research for the following parameters: (a) number of nodes, (b) size of quantum, (c) random time, (d) data transfer delay. In this paper node mobility were not taken into consideration. The scope of the network size was between 20 and 70 nodes. The minimal number of nodes was limited by network integrity. Maximum number of network nodes was the result of realistic networks scenarios and simulator computation limitation. We specify the quantum in relation to the nodes capacity. The scope of quantum size related to capacity was between 1 and 1/8. Our research focused on data
distribution phase, therefore, the process of building simple routing table known as first phase of the algorithm was not taken into account.

The following metrics were used in comparing the algorithm performance. Some of the chosen metrics is derived from ones suggested by IETF MANET working group [4):

- Number of control packets and data packets transmitted per data packet delivered

To investigate characteristics of data distribution service we provide high level metrics:

- **Succeed File Ratio**: The ratio of files size stored in the network which can be restored (sum of succeed and succeed conditional) versus all files size declared in simulation.
- **Succeed Quantum Ratio**: The ratio of file parts size stored in the network versus all file parts size declared in simulation.
- **Fail Quantum Ratio**: The ratio of file parts size that was broken versus all file parts size declared in simulation.
- **Unknown Quantum Ratio**: The ratio of file parts size that was delivered because network disruption versus all file parts size declared in simulation.
- **All nodes which store any quantum / all nodes**: the number of nodes that store any part file to the number of all nodes in the network.

**5. Results**

This section presents simulation results for different algorithm parameters. We analyze performance of the algorithm in the wide range of scenarios with variable quantum size and density of nodes. Each simulation for selected parameters, was repeated 30 times. The diagrams below presents the average values.

The efficiency of proposed algorithm in terms of Succeed File Ratio in function of number of nodes in the network was presented in Figure 2. For the small size of the network the value of the ratio is low because of network disruption. For the large size networks the ratio is falling because of increasing number of packets collision, what is observed especially for relatively small quantum/capacity (the number of data packets
transmitted is the highest). The highest efficiency was achieved for medium sizes of network and huge parts of the file (the biggest quantum is a cause of small number of packet transmitted).

![Figure 2. Algorithm efficiency graph of succeed file ratio in function of nodes number](chart)

In Figure 3 we have presented comparison of the ratios: succeed quantum, fail quantum and unknown quantum for quantum/capacity = 0.25. For the small sizes of network we observed 2% unlocated quantums (green curve). The reason is disjunctive network topology, what means file parts are stored at the injection point. For the small and huge sizes of network the number of fail quantums increase, the reason is the same as was specified for Figure 3. The avarage effectiveness of quantums allocation succees is larger than 70% in the network size larger than 20 nodes.
Figure 3. Algorithm performance for quantum/capacity = 0.25 in function of different network size

Figure 4 presents the ratio of number of nodes stored any quantum to number of all nodes in the network. The lowest value of the metric is the best and is achieved for quantum/capacity = 1. All curves decrease in function of increasing network size because the number of nodes necessary to all data allocation is constant.
6. Summary and conclusions

In this article a simple algorithm for the large-size data distribution and placement based on the tree-structure created on-demand has been presented. The authors describe two main phases necessary for the proper implementation of the algorithm: (a) data injection and (b) replication.

The present field of application is relatively young. Despite extensive study of literature it is difficult to find the same or even similar applications. Therefore, the results presented above can not be directly verified by comparison with other works. On the other hand, already early studies are quite promising. Our main goal was to determine the correct direction of future development. The presented results are promising, therefore, we can cautiously assume that the goal has been reached.

According to our best knowledge the algorithm is an effective way to solve the problem of the large-size data storage in Wireless Ad-Hoc Sensor Network environment.
7. References


