Load Balancing Routing Algorithm for Data Gathering Sensor Network

Evgeny Bakin, Grigory Evseev
State University of Aerospace Instrumentation
Saint-Petersburg, Russia
{jenyb, egs}@vu.spb.ru

Denis Dorum
The Hebrew University
Jerusalem, Israel
rarogf@yandex.ru

Abstract

Data gathering sensor network is considered in the paper. In these networks each node is used both as source of data and as retransmitter of packets, going from other nodes. A few important parameters of sensor network (such as lifetime, data rate, channel utilization etc.) depend on chosen routing scheme. Incorrectly chosen scheme results in appearance of bottle-necks, which can highly delay data transmission and lead to rapid battery discharge of particular nodes. In this paper we propose routing scheme for convergecast sensor networks, which decreases number of bottle-necks. Effectiveness of proposed algorithm is shown via simulation.

Index Terms: Sensor network, Convergecast, Data gathering, Routing, Load balancing.

I. Introduction

A. Sensor network description

Typical sensor network consists of a set of identical nodes, named "sensors", and a base station (BS) [1]. Sensor's purpose is measurement of particular physical parameters of controled target (e.g. temperature, humidity, vibration level etc.), preliminary processing of measurement results and transmission of appropriate messages by means of wireless channel. Messages, transmitted by sensors, are gathered on a base station. Thus logical structure of considered sensor networks is "'all-to-one"'. Process of messages gathering is usually called "'convergecast"' [2]. Let us denote a set of network sensors as $S = \{s_1, s_2, ..., s_N\}$, where N - number of sensors, and a base station as BS, or as s_0.

Every node can be both source of messages and a retransmitter of messages, going from other nodes. Thus despite the low radiated power of sensor transmitter, sensor network can cover large areas. In this case messages from distant sensors are delivered to BS through the chain of other sensors (see fig. 1, here $N = 14$, bold lines mark the routs of messages delivery from 10-th 14-th sensors).

The network can be described as graph of connectivity $\boldsymbol{G}(S, L)$, where S is a set of sensors and L is a set of links between sensors. The link $l_{i,j}$ presents in a set L if reliable communication between sensors s_i and s_j is arranged.

B. Convergecast mode of sensor network

In this paper networks with cyclic data gathering are considered. For such networks time can be split into *duty cycles* (DC) [4]. At the beginning of every DC each sensor generates message, which reflects condition of controlled target. During the other part of DC generated messages are delivered on BS (see fig. 2). Let l_i - length of rout, connecting sensor s_i and

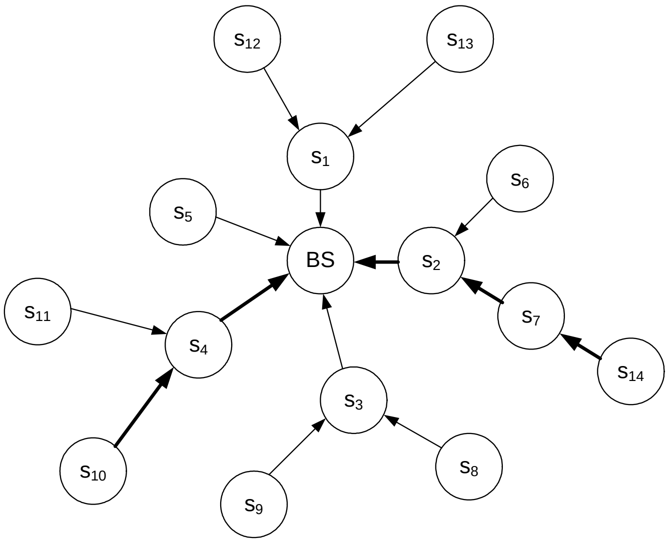

Fig. 1. Sample of sensor network

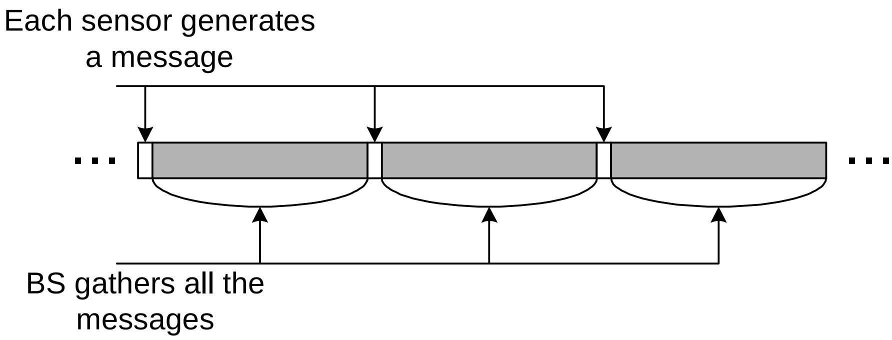

Fig. 2. Duty cycles

BS. Then during DC exactly $N_{ST} = \sum_{i=1}^{N} l_i$ successful transmissions are to be made. Let us denote this set of transmissions as P ($|P| = N_{ST}$).

Usually time of message generation is negligibly less than time of messages gathering. Obviously network operativeness is fully defined with DC duration, T. The less DC duration the more frequently information about controlled targets condition is updated at BS. DC duration depends much on chosen routing scheme. Thus, development of effective routing schemes for convergecast sensor networks is of a big interest.

II. Previous solutions

There are a lot of works devoted to problem of routing in packet wireless networks. Approaches and solutions used by authors vary much depending on optimization criterion [3]. Routing through the shortest path in graph $\boldsymbol{G}$ is frequently used. In such approach N_{ST}, the total number of transmissions in DC, is minimized. Thus total energy consumption of the network is minimized either. Obviously a few shortest paths connecting particular sensor and BS can exist. Then additional criteria are used.

In works [5]–[7] shortest paths are arranged so, that a set of links used in the routs would form a tree with a root at the BS. This sufficiently eases process of theoretical analysis but brings to unevenness of load distribution between neighboring sensors. For instance, let us consider a network, shown in figure 3. Here, if described approach is used, all the sensors with numbers ≥ 3 would transmit their messages only through sensor s_1, or only through sensor s_2. But, it is clear that it is more efficient to share the load equally between s_1 and s_2. Thus this method of routing is characterized with abundance of bottlenecks close to BS, which brings to DC duration increasing.

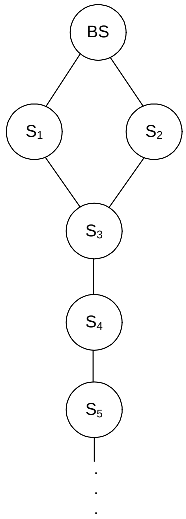

Fig. 3. Example of uneven load sharing among sensors

In [8] two heuristic load balancing routing algorithms are proposed. In first algorithm a term ”route bottleneck” is introduced. This term denotes a particular sensor which takes part in maximum number of retransmissions during DC. Making a rout from the next sensor to BS, algorithm chooses that rout from a full set of possible routs which bottleneck has a minimal load. This algorithm has quite high complexity because needs full search over all possible routs for every sensor.

In the next algorithm a term ”cost for transition” is introduced. At the beginning of the algorithm work costs for all the sensors are equal. Routs to BS are arranged one by one for every sensor. The rout which total cost is minimal is chosen. After that costs of all the sensors in a chosen rout are to be increased on one. For the next sensor rout is arranged accounting updated costs etc. Similar algorithm is also given in [4]. This algorithm has low complexity but in this way messages can be delivered not through the shortest path. This increases total energy consumption of the network.

III. Proposed algorithm description

As was mentioned above, the main strategy is routing through the shortest path. This allows minimizing total energy consumption of the network. As additional criterion of optimization load balancing is proposed. The sensors, which distances from BS are equal, have to be equally loaded during DC. The proposed routing algorithm is based on classic Dijkstra algorithm.

A. Classic Dijkstra algorithm (CDA)

Dijkstra algorithm is a particular case of dynamic programming algorithm. In CDA for each sensor the *metrics* d_i is assigned. Initially all sensors metrics are equal to infinity and

BS metrics is equal to zero. A set of considered sensors U is introduced (initially empty). In each iteration CDA chooses sensor, which has minimal metrics and doesn't belong to U. All the neighbors of chosen sensor are subsequently considered. If any neighbor's metrics excesses over chosen sensor's metrics on a value bigger than weight of their connecting link, than current routs and current metrics are modified (see pseudocode of CDA below). After all neighbors are considered, the chosen sensor is placed in U. For CDA description on pseudocode we introduce the following notation:
U - set of already considered sensors;
d_i - sensor s_i metrics ($i = \overline{1, N}$);
r_i - ordered subset of links from L, keeping the shortest rout from BS to s_i ($i = \overline{0, N}$);
$r_k = [r_j, l_{k,j}]$ - procedure of addition of $l_{k,j}$ to the end of set r_j and it's assignment to r_k.

Algorithm 1 Classic Dijkstra algorithm of finding shortest rout between BS and sensor s_t

$d_0 \leftarrow 0$, $r_0 \leftarrow 0$
$U \leftarrow \oslash$
for $i = \overline{1, N}$ **do**
  $d_i \leftarrow \infty$
end for
while $s_t \notin U$ **do**
  Let $s_j \notin U$ - sensor with minimal metrics d_j
  for $\forall s_k \notin U$, so, that $l_{k,j} \in L$ **do**
    if $d_k > d_j + 1$ **then**
      $d_k \leftarrow d_j + 1$
      $r_k \leftarrow [r_j, l_{k,j}]$
    end if
  end for
  $U \leftarrow U \cup s_j$
end while

After finishing, the ordered set r_t keeps the shortest routs between BS and sensor s_t. Let us denote the described function of finding a rout as $r_k = \text{Dijkstra}(s_k)$. Now, basing on CDA we represent an algorithm which allows balancing of loads either.

B. Modified Dijkstra algorithm (MDA)

The main idea of proposed algorithm is in dynamic adjustment of links weights in process of routing to base station. In proposed algorithm sensors are chosen one by one. For the chosen sensor rout is calculated via CDA. After that, weights of all the links, used in the rout are to be increased on $\frac{1}{N^2}$. This value is chosen so, because total length of the rout must not increase on a value more than 1 during algorithm fulfillment (otherwise, the next rout can go not through the shortest path). Thus, the more messages are transmitted through the link, the more cost for it's usage.

For algorithm description in pseudocode we introduce the following notation:
$w_{i,j}$ - weight of link $l_{i,j}$, connecting sensors s_i and s_j.

As one can see, proposed algorithm MDA is still polynomial and it's complexity practically doesn't excess complexity of CDA.

Algorithm 2 Modified Dijkstra algorithm of finding shortest rout between all the sensors and BS

$\forall w \leftarrow 1$
for $k = 1, 2, \cdots N$ **do**
  $r_k \leftarrow$ Dijkstra(s_k)
  for $\forall l_{i,j} \in r_k$ **do**
    $w_{i,j} \leftarrow w_{i,j} + 1/N^2$
  end for
end for

IV. Simulation results

To study effectiveness of proposed algorithm the following simulation was performed:

1) Random network graph of a given size N was generated with algorithm MIN-DPA, described in [9].
2) For the generated network routs were formed with CDA and MDA.
3) For both cases duty cycle was calculated using algorithm [10].

The described procedure was repeated 10000 times for each N. Average DC durations for CDA and MDA were estimated (see fig. 4).

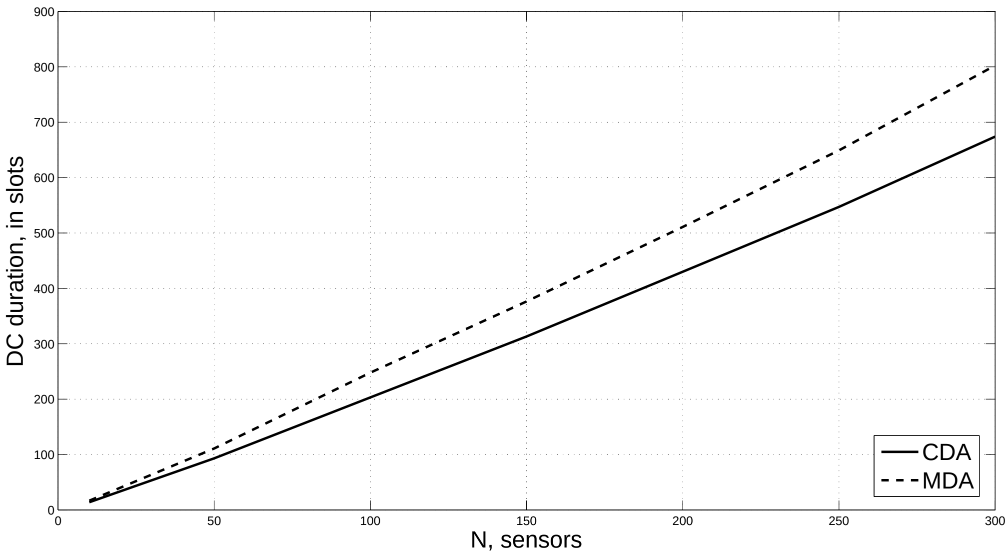

Fig. 4. Simulation results

V. Conclusion

In the paper simple routing algorithm was proposed. The algorithm balances the load between sensors, thus minimizing number of bottlenecks in a network. It's complexity polynomial and practically doesn't excess complexity of classic Dijkstra algorithm. Simulation shows that usage of proposed algorithm allows minimizing of duty cycle duration for up to 20%.

REFERENCES

[1] K. Sohraby, D. Minoli, and T.Znati, ”Wireless Sensor Networks: Technology, Protocols, and Applications,” *Wiley-Interscience*, 2007.
[2] F. Schill, U.R. Zimmer, J. Trumpf, ”Towards optimal TDMAs scheduling for robotic swarm communication,” *Proceedings of the TAROS international conference*, p. 215, 2005.
[3] A. Ephremides, S. Verdu, ”Control and optimization methods in communication networks problems,” *IEEE Transactions on Automatic Control*, vol. 34(9), pp.930-942, 1989.
[4] Y.-F. Wen, F.Y.-S. Lin, H.-S.Wang, ”A TDMA-based scheduling and routing algorithm for data-centric wireless sensor networks,” *Proceedings of IEEE, GLOBECOM07*, 2007.
[5] S.C. Ergen, P. Varaiya, ”TDMA scheduling algorithms for wireless sensor networks,” *Wireless Networks, Springer*, pp.985997, 2009.
[6] A. Berfeld, D. Mosse, ”Efficient scheduling for sensor networks,” *Proceedings of the 1st International workshop on Advances in Sensor Networks(IWASN06)*, 2006.
[7] C. Renner, V. Turau, C. Weyer, ”Performance of energy efficient TDMA schemes in data gathering scenarios with periodic sources,” *Proceedings of the 7th IEEE International Conference on Networked Society Systems (INSS10)*, 2010.
[8] D.J. Vergados, K. Maria, ”Optimizing end-to-end TDMA scheduling in ad-hoc networks on random topologies”, *The 6th Annual Mediterranean Ad Hoc Networking Workshop*, vol. 71-78, 2007.
[9] F.A. Onat, I. Stojmenovic, H. Yanikomeroglu, ”Generating random graphs for simulation of wireless ad hoc, actuator, sensor, and wireless networks,” *Pervasive and Mobile Computing*, vol. 4, pp. 597615, 2008.
[10] E. Bakin, ”Algorithm of schedule calculation for centralized sensor network,” *Proceedings of the International Forum Modern information society formation problems perspectives, innovation approaches*, p. 112, 2010.