This is a draft of an article published in
Kirt, T. (2008). Fast and Adaptive Scheduling in Ad Hoc Networks. In Proceedings of the 11th biennial Baltic electronics conference (pp. 251 - 254). Tallinn, Estonia: Tallinn University of Technology.

Fast and Adaptive Scheduling in Ad Hoc Networks

Toomas Kirt

Institute of Cybernetics at TUT, Akadeemia tee 21, 12618 Tallinn, Estonia, E-mail: Toomas.Kirt@mail.ee

ABSTRACT: In several applications of ad hoc networks an effective scheduling algorithm is needed. Scheduling is essential for optimizing power consumption and increasing transmission capacity of the network. A schedule of operations performed by the network nodes can be formed by solving the graph vertex colouring problem. To solve the graph colouring problem in the environment of ad hoc networks is complicated, because an ad hoc network is dynamic and only local information can be used. In this paper a graph colouring algorithm is proposed that meets the requirements of an ad hoc network. The algorithm consists of two parts. The first part uses a greedy approach to find a suboptimal solution and the second part is a self-organizing part that optimizes the schedule and adapts with the changing environment. The proposed algorithm is fast and also fail-safe because of its simplicity in the origin.

1. Introduction

In some applications of ad hoc networks it is needed to agree a schedule so that two neighbouring nodes do not perform the same operation at the same time. For example, a data gathering network of Radio Frequency Identification (RFID) tags readers where two neighbouring readers can not read a tag at the same time because of signal interference. The readers have to arrange in which order they read the tags. As an ad hoc network does not have any central station that manages the network only local information, available from the immediate neighbourhood of a node, can be used. It makes the problem much more complicated and some traditional approaches are not applicable. In this paper an algorithm is proposed that satisfies the constraints caused by the origin of an ad hoc network.

The scheduling problem is reducible to the graph vertex colouring problem that is a NP-hard (nondeterministic polynomial time-hard) problem [1]. To solve the problem graph vertexes should be coloured so that the two vertexes having a common edge do not have the same colour. Usually a heuristic approach is used to find a suboptimal solution within an acceptable time range. A variety of algorithms have been proposed to solve the graph colouring problem heuristically. The most elementary one is a greedy algorithm that is quick and provides quite good solutions [2]. Another suitable solution is a self-organizing approach that uses only local information and is applicable to the ad hoc environment [3].

The self-organization is defined as a spontaneous creation of a globally coherent pattern out of local interactions between initially independent components [4]. The main characteristics of self-organization are as follows: increase in the order or organization of the system; absence of any external control; adaptability in the presence of perturbations and change; and being a process [5]. An ad hoc wireless network can be considered as a self-organizing system as it is maintained by using only local interactions.

An ad hoc wireless network consists of nodes that are independent, work in a similar manner, and do not depend on a central station that manages connections between them or controls them. The nodes in an ad hoc network usually use a common optimal power level that is the lowest transmitting power level that maintains the network connectivity [6]. The nodes are directly connected only to their neighbours that are in a wireless transmission range and indirectly to others, relying on its neighbours to forward messages towards the destination [7]. An ad hoc network builds its structure autonomously and reacts to changes in the structure when a node joins, moves around or leaves. The network nodes share information locally only with their neighbours, but despite the fact the information spreads through the network rather rapidly. It is in accordance with the idea of the small-world network [8].

The aim of this paper is to propose a method of graph colouring that can be used in the dynamic environment where only local information from the immediate neighbouring nodes is available. The method must be fast and adaptive and have ability to find an optimal solution for graph colouring problem and also to find solution for distributing the information about the number of colours. Many solutions have been proposed but usually they require some global information i.e., information about all the vertexes or a global approach to solve the graph colouring problem and can not be used in a distributed environment.

As the ad hoc wireless network is an active research area, several researchers have tackled the problems of scheduling and graph colouring. As the graph colouring is a complex problem (especially in the ad hoc networks) it is quite often left out from the study and some other simpler solutions are used, like to use predefined schedules that can be applied when necessary [9] [10]. In several applications the graph colouring problem is solved by using some heuristic method, but usually a centralized approach is used [11] or the number of colours is predefined [12]. The most suitable seems to be the solution where the dsatur algorithm is used in a distributed manner [13], but as the results of simulations show the method takes too many steps to find a solution. The slowness is also a problem of a self-organizing approach to solve the graph colouring problem [3] although the method works in a distributed way.

As there is not available any method that satisfies all constraints of our task a new combined method is proposed. The proposed method colours a node, finds the number of colours needed to colour the network, distributes the number of colours across the network when the number is increased or decreased.

The paper is divided into two parts. The combined method is introduced in the first part of the paper and in the second part the results of the performed simulations are presented and discussed.

2. Method

The ad hoc network can be regarded as an undirected graph G = (V, E), where V = $\{v_1, ..., v_n\}$ is a set of *n* nodes or vertices and E = $\{e_1, ..., e_m\}$ is a set of *m* connections or edges. For colouring the graph we have to find a colour for each vertex from the set of *k* available colours C = $\{c_1, ..., c_k\}$so that the colour of a node does not match with the colour of the neighbouring nodes.

The proposed method is combination of two colouring algorithms and additionally the self-organizing approach is used to distribute the number of colours among the network nodes. The first colouring method uses a greedy algorithm to perform initial and fast colouring and the second one is a self-organizing algorithm to adapt with the changing environment and to move closer to the optimal colouring. Both methods are very simple ones and are computationally effective and do not cause too much additional network traffic.

The greedy algorithm is fast and finds reasonably good solutions. The greedy algorithm works as follows:

- find the lowest available colour *c* that is not used by the neighbouring nodes and colour the node;
- if all the colours are used and the node is not coloured then increase the number of colours *k* = *k* + 1 and colour the node with *k*.

The self-organizing approach for colouring the graph is based on a stochastic and self-organizing solution to solve complex problems [14]. In his paper an extended version of the method is used (see [3]). The key concepts of the method are the evaluation function to measure constraint satisfaction and the fitness condition to assess whether to change the colour of a node.

The evaluation function O between two vertexes v_1 and v_2 is defined in case of the colouring problem – the value is 1 if the constraint between v_1 and v_2 is satisfied and it is 0 otherwise and formally as follows:

$O(v_1, v_2) = 1$ if $v_1.colour \neq v_2.colour$

0 otherwise

Reaction takes place between two connected vertexes v_1 and v_2 if the following fitness condition is satisfied:

$$O(v_1, v_2) \leq O(v_1', v_2).$$

The self-organizing graph colouring algorithm consists of following steps:

- calculate the evaluation function O for the current colour *c*;
- select randomly a new colour c' within the range of 1...*k*;
- calculate the evaluation function O' for the new colour *c*';
- if the fitness condition O ≤ O' is satisfied then apply the new colour *c*'.

An example of choosing a new colour is given in Figure 1. The value of the evaluation function O before the reaction is in Figure 1a. If the new colour is 2 the fitness condition is not satisfied and there is no reaction, see in Figure 1b. If the new colour is 3 the fitness condition is satisfied and the reaction occurs, see in Figure 1c.

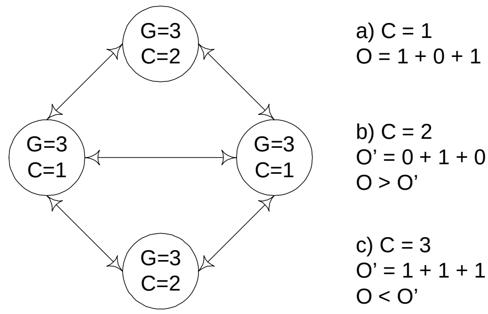


Figure 1. The selection of a new colour of a node. The value of the fitness function before (a) and after (b, c) selecting a new colour.

An important part of the method is messaging between the nodes. The network is an ad hoc network, it means that there is no central station that controls the communication between the nodes. All the nodes can communicate directly with the neighbouring nodes within a transmission range. A node sends immediately a broadcast message if it has received a message about the new number of colours or the node has changed a colour or has increased or decreased the number of colours. All messages contain information about the colour of a node *c*, the number of colours *k* and the unique message ID to avoid propagation of the same messages around the network.

Information about the number of colours is spread across the network in a self-organizing way, it means that the information is sent only to the immediate neighbours. It follows the idea of a small world phenomenon and the self-organizing behaviour of an ad hoc network it is examined in several experiments [15]. If the number of colours is increased or decreased then such information spreads as a wave throughout the network.

The general algorithm consists of four phases (see

Figure 2). The initial phase is used to get to know who the neighbours are and what their colour is. Initially the colour of a node is 1. When a node joins with a network it immediately sends a broadcast message. All the nodes in the transmission range that hear the message send a message back with their colouring information and the node moves to the second phase.

The second phase is a greedy one. The node initiates a time-out counter and if the counters value is 0 the greedy algorithm is initiated to colour the node. If a node gets feedback that the selected colour is not in a conflict with the colours of its neighbours the node turns into the self-organizing phase.

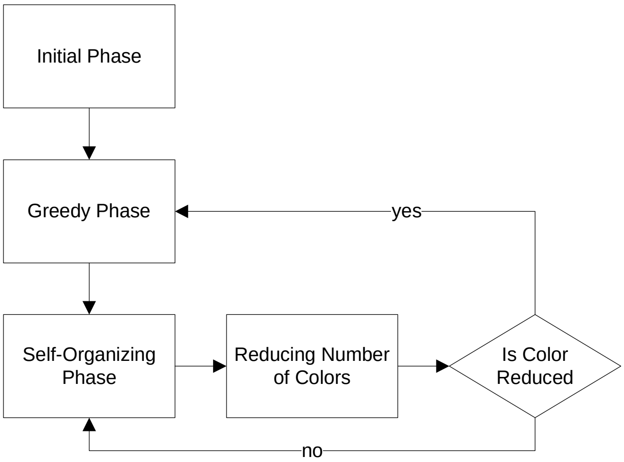


Figure 2. The structure of the colouring method

In the third phase the self-organizing algorithm is used to recolour the graph. In this experiment the fitness condition is defined as follows:

(O < O' or O' = 0) and $c \neq c'$.

The fitness condition prefers better colouring and recolouring if the value of the colouring remains correct (all the neighbours are coloured differently). The colour is not changed if a new colour is the same as the old one for avoiding the needless network traffic.

If a node is coloured by the colour k finds a new colour $c < k$ then it enters to the phase of reducing the number of colours. It is crucial part of the algorithm, because if the greedy algorithm finds a suboptimal solution or a node, coloured by a colour k, leaves from the network, then it is needed to reduce the number of colours. After selecting the new colour the node reduces the number of colours $k = k - 1$ and sends a new k to the neighbours. If a neighbouring node receives the message it reduces the number of colours and forwards the message.

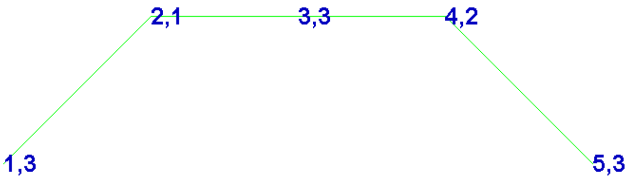
3. Results

To measure the adequacy of the method a simulation environment was built and several experiments were performed. The simulation environment is realized in Matlab and the results are visualized by using Matlab graphical tools. The simulation uses discrete time and is step based. At each step nodes receive messages from the neighbouring nodes in the transmission range, perform calculation to apply colouring and send messages to the neighbours.

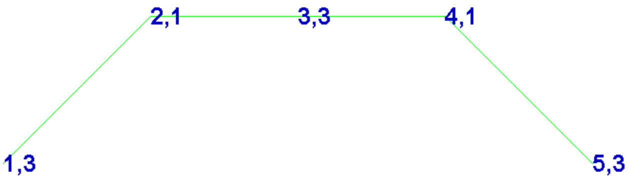
Table 1. Results of the simulations

Exp.	No of nodes	No of colours Min	Mean	Max	No of steps Min	Mean	Max
A	100	13	15.1	17	51	73.5	115
B	100	14	15.1	17	11	11.2	15
C	250	15	16.4	18	53	65.6	81
D	250	15	16.1	17	11	11.3	17
E	250	14	14.5	16	101	101	101

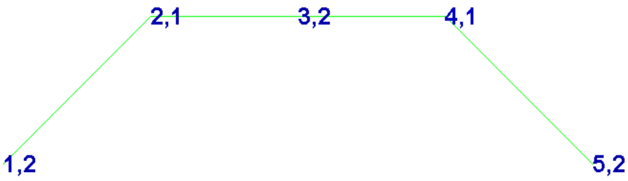
The results of simulations are given in Table 1. In the experiments “A” and “C” the self-organizing method [3] and in the “B”,“D”, and “E” the new combined method was used. Two graphs were used (consisting of 100 and 250 nodes) and the optimal number of colours on both cases was 13. The algorithm sopped when all the nodes are coloured correctly except the experiment “E”, where 30 additional steps were performed after the greedy phase. The results indicate that the number of steps does not depend on the number of nodes, the new method had a better performance and the number of colours became lower when the algorithm was kept running longer.



a) Step 9



b) Step 10



c) Step 13

Figure 3. Adaptability of the graph colouring method (legend: number of a node, colour)

The main challenge of the colouring is to avoid the number of colours to fluctuate as there could simultaneously start to spread two controversial messages –one of which tries to reduce and the other one to increase the number of colours. To avoid such fluctuations all the messages of reducing the number of colours are marked by unique id and so the repetition of the same message is avoided. To avoid the incorrect colouring when a node receives a message that requires to reduce the number of colours and if it is coloured with the colour *k* then the node turns immediately to the greedy phase. If a node is not able to find a free colour then it does not reduce the number of colours and cancels the transmission the message of the reduced number of colours and instead it is transmitting a message containing the previous number of colours. Such approach makes the network more stable.

Adaptability of the algorithm is visible in Figure 3. When the greedy colouring is applied the number of colours turned to be higher than the optimal solution would be (Figure 3a). As the self-organizing way to colour a graph used random selection of colours there is possibility to find a configuration where an obstructive node changed the colour and makes it possible to find an optimal solution (Figure 3b). After a several steps nodes 3 and 5 selected randomly a new colour that allowed to reduce the number of colours *k* by one (Figure 3c).

As a result we could see that the colouring was performed fast and accurately. When a node is correctly coloured in the greedy phase it maintains the accurate colouring if there is not any node moving or joining the network. If some perturbations happen the self-organizing recolouring is applied.

4. Conclusion

In this paper a method is proposed for solving the scheduling problem in the environment of ad hoc networks. To agree a schedule so that two neighbouring nodes do not perform the same operation at the same time the graph colouring problem must be solved. The proposed solution satisfies all the restrictions specific to the ad hoc networks. The algorithm is fast, fully distributed, adaptive and has ability to move towards an optimal solution of the colouring problem. The results of the performed simulations indicated that the combination of two colouring methods gives a better result than the methods applied separately and some simple local rules and self-organization can be used to solve complex optimization problems.

References

[1] M. R. Garey, and D. S. Johnson, “Computers and Intractability: A Guide to the Theory of NPcompleteness”, Freeman, San Francisco, 2003.

[2] J. C. Culberson, “Iterated greedy graph coloring and the difficulty landscape”, Technical Report 9207, Department of Computing Science, The University of Alberta, Edmonton, Alberta, Canada, June 1992.

[3] T. Kirt, “Graph coloring by self-organizing algorithm”. International Transactions on Systems Science and Applications, Special Issues on "Self-Organizing, Self-Managing Computing and Communications", vol. 2, No. 3, pp. 309- 314, 2006.

[4] F. Heylighen, “The Science of Self-organization and Adaptivity,” in The Encyclopedia of Life Support Systems, L. D. Kiel, Ed., 2001, Oxford, Eolss Publishers, 2001 Available: http://www.eolss.net.

[5] T. De Wolf, and T. Holvoet, “Emergence Versus Self-Organisation: Different Concepts but Promising When Combined”, in Engineering Self Organising Systems: Methodologies and Applications, S. Brueckner, G. Di Marzo Serugendo, A. Karageorgos, and R. Nagpal, Eds., Lecture Notes in Computer Science, vol. 3464, Springer, Berlin, 2005, pp. 1- 15.

[6] V. Kawadia, and P. R. Kumar, “Principles and Protocols for Power Control in Ad Hoc Networks,” IEEE Journal on Selected Areas in Communications, vol. 23, No. 5, pp. 76- 88, 2005.

[7] L. M. Feeney, “Energy efficient communication in ad hoc networks,” in Mobile Ad Hoc Networking, S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, Eds. Wiley-IEEE Press, 2004, Ch. 11.

[8] D. J. Watts, and S. H. Strogatz, “Collective Dynamics of 'Small-World' Networks”, Nature, vol. 393, pp. 440- 442, 1998.

[9] A. Ruzzelli, R. Tynan, and G. M. P. O'Hare, “A Low-Latency Routing Protocol for Wireless Sensor Networks”, in Proceedings of Advanced Industrial Conference on Wireless Technologies (ICW 2005, Montreal, Canada, August 2005), IEEE Press.

[10] A. Ruzzelli, G. M. P. O'Hare, M. J. O'Grady, and R. Tynan, “Adaptive Scheduling in Wireless Sensor Networks”, in Proceedings of 2nd IFIP International Workshop on Autonomic Communication (WAC 2005, Vouliagmeni, Athens, October 2005), Lecture Notes in Computer Science (LNCS), Springer-Verlag.

[11] H. Lee, and C.- C. Lee, “An Integrated Multi-hop Cellular Data Network,” in Proceedings Vehicular Technology Conference, October 2003, Orlando, FL.

[12] L. Hu, “Distributed code assignments for CDMA packet radio networks”, IEEE/ACM Transactions on Networking, vol. 1, No. 6, pp. 668- 677, 1993.

[13] R. Battiti, A. Bertossi, and M. Brunato, “Distributed Code Assignment in Multihop Radio Networks: Object-Oriented Software Simulations”, in Proceedings of SoftCOM 2000 (Rijeka, Croatia, October 2000).

[14] Y. Kanada, and M. Hirokawa, “Stochastic Problem Solving by Local Computation Based on Self-organization Paradigm,” in Proceedings of 27th Hawaii International Conference on System Sciences, R. H. Lathrop Ed., Wailea, HI: IEEE Computer Society Press, 1994, pp. 82- 91.

[15] T. Kirt, and A. Anier, Self-organization in ad hoc networks, In Proceedings of the 10th biennial Baltic electronics conference, T. Rang Ed., Tallinn, Estonia: Tallinn University of Technology, 2006, pp. 149- 152.