Binary Consensus-Based Decentralized Algorithm for Event Detection in Large Scale Monitoring Systems

CRISTIAN DRĂGANA, GRIGORE STAMATESCU, DAN POPESCU
Department of Automatic Control and Industrial Informatics
University Politehnica of Bucharest
313 Splaiul Independenței, 060042 Bucharest
ROMANIA
cdragana@imtt.pub.ro

Abstract: - Decentralized sensor fusion plays a particularly important role in event occurrence detection when it comes to large scale monitoring systems. Regarding WSNs, a decentralized sensor fusion mechanism provides efficient information extraction and data reduction by means of reducing the energy and communication constraints of the embedded sensing nodes. In large scale monitoring systems event occurrence detection can be achieved by different methods based on decentralized detection. In some cases such as earthquake detection, a binary sensing approach is proper. This mainly concerns to binary consensus algorithms intended to allow efficient in-network data processing. The paper discusses a binary consensus mechanism for use in wireless sensor networks in large scale monitoring systems for event occurrence detection. We focus more on real hardware and software implementation instead of theoretical work. Experimental results such as convergence time are provided stemming from a flexible testbed based on Contiki-Cooja WSN simulator. Moreover, we are able to perform energy consumption estimation for each sensing node by means of convergence steps and power trace. Experimental results showed that time convergence can vary in a rather unexpected way for different number of sensor nodes, while number of convergence steps and power consumption are linear dependent to the number of sensor nodes. A network architecture based on clusters of nodes should be considered for wireless sensor networks consisting of large number of sensor nodes.

Key-Words: - wireless sensor networks; decentralized sensor fusion; binary consensus; Contiki - Cooja

1 Introduction

In recent years plenty of studies focused on in-network data processing have leaded to the emergence of efficient decentralized sensor fusion algorithms designed for networks of intelligent sensors, more often referred to as wireless sensor networks (WSN). These mechanisms provide solutions for consensus decision making in a way that leverages the on-board computing resources and reduces the burden on the communication channel.

This approach promises efficient information extraction and data reduction by means of reducing the energy and communication constraints of the embedded sensing nodes. In a consensus decision making process the agents try to agree upon a certain value, state or decision, in the best interest of the whole. In a decentralized approach based on in-network data processing, the network agents seek to reach decision using only local information exchange (data exchange with neighbor agents). In some cases, the consensus problem may be limited to settle whether a statement is true or false in a binary sensing approach. Based on binary sensing, each network agent has to decide if an event has occurred or not. For example, in an underground parking space, ventilation system can be controlled depending on the concentration of contaminants in the air. Each network agent measures the air quality and decides whether the level of contaminants is bellow or over the allowed limit, then the network agents seek to reach a common decision. In this way the ventilation level can be increased if an exceeding of the allowed limit has occurred. This basic example of binary sensing approach can be extended to large scale monitoring systems such as an earthquake detection system or other critical monitoring systems.

Existing work on consensus problem usually focuses on theoretical research and ideal simulation of the algorithms without considering the communication constraints and other limitations of the wireless sensor networks.

Nowadays, wireless sensor networks have become a robust technology which enables reliable deployments. Reduced size electronic parts with increased computation capabilities and energy efficiency are available for advanced embedded systems. However, given the limited computing resources and communication constraints, accurate contributions focused more on real hardware and software implementation of consensus algorithms are increasingly desired.

Moreover, current available virtual tools for simulation and prototyping enable researchers to improve and speed up development and lower its cost [1]. Available full system environments, offering flexible virtual network testbed, are usually associated to two major platforms for WSN implementation, Contiki and TinyOS.

Contiki [2] is a state-of-the-art, open source operating system for Wireless Sensor Networks, Internet of Things (IoT) and other network embedded devices.

In [3] a binary consensus algorithm is implemented and tested in a set of TinyOS based sensor nodes. Convergence time is evaluated for different network topologies, number of motes and network density.

In [4] an algorithm that reduces the multivalued consensus to the binary consensus problem for asynchronous distributed systems is submitted.

An evaluation of convergence rates of binary majority consensus algorithms in networks with different types of disturbances is described in [5] and studies the potential capacity of randomization to foster convergence.

In [6] a general architecture of a large scale heterogeneous monitoring system and information processing by means of a decentralized fusion approach is discussed. The monitoring system consists of surveillance UAV (unmanned aerial vehicle), ground sensor network subsystems and a ground control center. Information extraction from ground sensor network can be achieved by means of a binary consensus approach.

The main contribution of this work is to provide a binary consensus algorithm adjusted for event detection in large scale monitoring systems based on a real WSN architecture. We take advantage of the Contiki/COOJA simulation environment in order to test the performance of the algorithm on a WSN testbed comprising Tmote Sky sensor nodes. Energy consumption estimation for each sensor node, by means of convergence steps and power trace function, is carried out for different scenarios. A flexible and robust node to node communication mechanism has been implemented in a random neighbor communication approach. Our main view is that this type of approach would easily apply to other consensus decision making algorithms.

The rest of the paper is structured as follows. Section 2 contains a brief overview of binary consensus and discusses the design and implementation of the adjusted binary consensus algorithm for Contiki OS based sensor nodes. Section 3 showcases experimental results drawn from Contiki/COOJA simulations. Last section concludes the paper and highlights future work directions.

2 Methods and Algorithms

Binary consensus algorithm for event occurrence in WSN is based on a binary sensing approach for sensor nodes. Each sensor node decide whether an event occurred or not, and marks its state. The convergence step allows each node to discover neighbor state and updates its own state based on a state updating rule. In this way, all the sensor nodes achieve the state held by majority, and consensus is reached.

Given these, at any convergence step a sensor node may be in one of the states described in Table 1. The state updating rule is called when two sensor nodes discover their current states. State updating rule, as quoted from [7] and also used in [3] is described in Table 2. When both sensor nodes have the same state they keep the current state unchanged. This case is not specified in Table 2.

Table 1 Possible sensor node states

State	Meaning
0	Majority opinion is FALSE
1	Majority opinion might be FALSE
2	Majority opinion might be TRUE
3	Majority opinion is TRUE

When all the sensor nodes believe that majority opinion is FALSE or might be FALSE (state is 0 or 1), it means the event has not occurred. Similar, if all the sensor nodes believe the majority opinion is TRUE or might be TRUE, then the event has occurred.

Table 2 State updating rule

Current states pair	New states pair
(0,1)	(1,0)
(0,2)	(1,0)
(0,3)	(2,1)
(1,2)	(2,1)
(1,3)	(3,2)
(2,3)	(3,2)

In a real WSN, a decentralized event detection mechanism based on the binary consensus algorithm should first consider the main challenge: communication problem. How does a sensor node find a partner and decide to share state information

ISBN: 978-1-61804-355-9

with? How does a sensor node decide when to stop communicating?

To complete this challenge a random neighbor communication approach based on broadcast and unicast communication was implemented. Assuming that the sensor node has decided on its initial state, communication process is activated and it follows the state machine transition rule illustrated in Fig.1 and further detailed.

When communication process starts the sensor node goes to *Idle* step. A random timer is started. If during this time a broadcast message from a neighbor consisting of identification ID and current state is received, the sensor node goes to *Unicast M1*step. If not, it goes to *Broadcast* step.

In *Unicast M1*step, the sensor node responds to its neighbor with a unicast message consisting of its identification ID and current state, and waits for a confirmation. A timer is started. If during this time the sensor node receives a unicast message from its neighbor with a confirmation code, it goes to *Update* step. If not, it goes back in *Idle* step.

In *Update* step, the sensor node updates its current state according to the state updating rule, and goes back in *Idle* state.

In *Broadcast* step, the sensor node broadcasts a message consisting of identification ID and current state. A timer is started. If during this time the sensor node receives a unicast message, it goes to *Unicast M2 Update* step. If the timer expires, it goes back in *Idle* step.

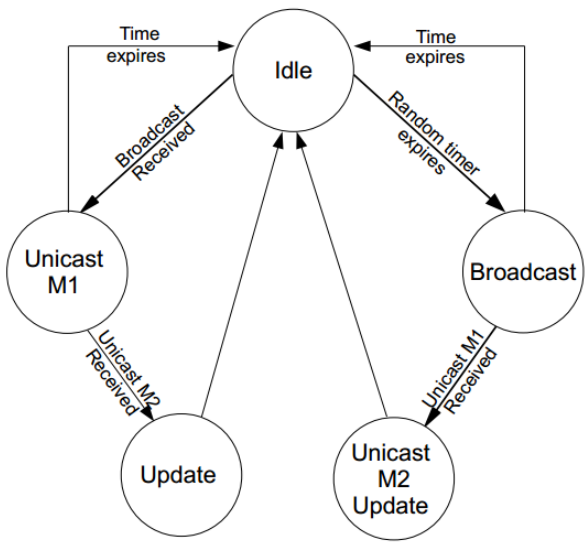

Fig.1 Stage machine transition of communication mechanism

In *Unicast M2 Update* step, the sensor node sends a unicast message consisting of identification ID and a confirmation code and updates its current ID according to the state updating rule. Then it goes back to *Idle* step.

During this time, when the sensor node goes to *Idle* step, the new state is compared with the last one. If the last state was close to TRUE and the new one is still close to TRUE a value is incremented. If not, the value is reset to zero and similar for state close to FALSE. When the value is equal to a constant parameter the communication process is deactivated.

In order to test the event detection mechanism on a real WSN system, the binary consensus algorithm was implemented based on Contiki OS for Tmote Sky sensor node. It has the main specifications listed in Table 3.

The implemented algorithm assumes that the sensor node has already decided on whether the event has occurred or not and it starts with a preset state value. Given this, two types of Tmote Sky sensor nodes were defined in Contiki/COOJA simulation environment: #Sky_true – sensor nodes that start with state *3* and #Sky_false – sensor nodes starting with state *0.*

Table 3 Tmote Sky main specifications

Tmote Sky	
CPU	8Mz TI MSP430F1611 (10k RAM, 48k Flash)
RF	250kbps 2.4GHz IEEE 802.15.4 Chipcon Wireless Transceiver
Supply Voltage	2.1 to 3.6V
Onboard sensors	Humidity, Temperature, and Light sensors
Signal range	50m indoors 125m outdoors

Regarding performance analysis, convergence time, number of convergence steps and power consumption are evaluated. Among these, power consumption requires an online energy estimation mechanism implementation. The Contiki OS has an embedded *powertrace* function [8] that allows us to know the time spent in the following states: CPU (active), LMP (Low Power Mode), Transmit and Listen. A linear model for sensor node energy consumption estimation was used. Total energy consumption E is defined as fallowing:

$$E = (I_m \cdot t_m + I_l \cdot t_l + I_t \cdot t_t + I_r \cdot t_r) \cdot V \quad (1)$$

where I_m is the current draw of the microprocessor active, t_m is the time during the microprocessor has

ISBN: 978-1-61804-355-9

been active, I_l is the current draw of the microprocessor in low power mode, t_t is the time during the microprocessor has been in low power mode, I_t is the current draw of the transmission device in transmit mode and t_t is the transmitting time, I_r is the current draw of the transmission device in receive mode and t_r is the receiving time, V is the supply voltage.

Time parameters in equation (1) are returned by the *powertrace* function. The current consumptions of the CPU and radio transceiver are available in sensor nodes datasheets. Tmote Sky current consumptions according to [9] are listed in Table 4.

Table 4 Tmode Sky Current Consumptions

CPU (active) current at 3Vcc, 1MHz	500	µA
CPU (LPM) current at 3Vcc, 33kHz	3	µA
Radio Rx current	22	mA
Radio Tx current	19	mA

4 Experimental Results

To analyze the convergence time, convergence steps and power consumption in particular, the binary consensus algorithm was tested on different size WSNs comprising Tmote Sky sensor nodes using the Contiki/Cooja Simulation Environment. The code executed by the emulated sensor nodes in the simulation is the exact same code that could run on physical Tmote Sky nodes. In Fig. 2 is illustrated an example of a network topology used for algorithm testing and evaluation.

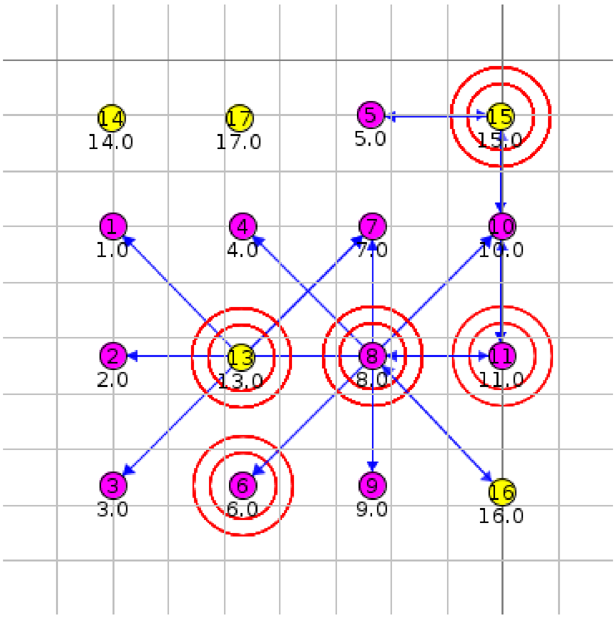

Fig.2 Example of network topology used

As mentioned in previous section, sensor nodes start with a preset state value. Yellow nodes are #Sky_true sensor nodes, while #Sky_false sensor nodes are marked with magenta color. As one can see, sensor nodes were distributed in a x,y grid topology, keeping a constant distance of 20m between them, with a transmission range of 50m. A rate of about 30% of the sensor nodes start with TRUE state, while the rest begin with FALSE state.

For the tested WSN topology in Fig.2, an average convergence time of almost 12 seconds was needed.

It took an average of 55 convergence steps to agree that the event has not occurred. Based on the online energy estimation mechanism an average of 6.8mW power consumption was calculated for a convergence step. Based on these results, each sensor node has an estimated power consumption about 400mW. Note that this does not include power consumption of the sensor node in idle state.

Power consumption evolution during consensus algorithm execution of a randomly chosen sensor node is illustrated in Fig.3.

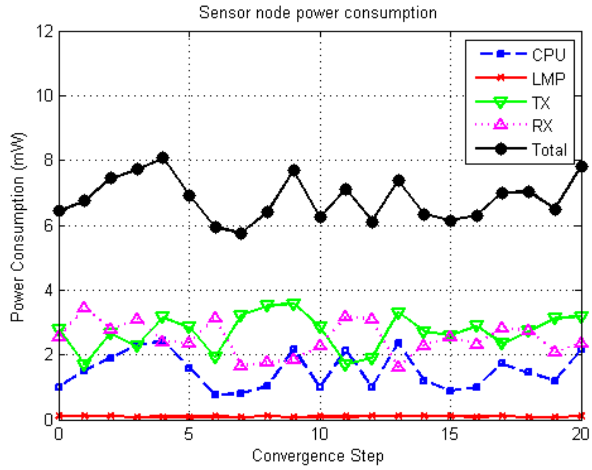

Fig.3 Sensor node power consumption during algorithm execution

As one can see, LMP power consumption is the lowest, while RX and TX processes require most of the energy. It is well known that transmitting and receiving represent the main energy consuming processes. TX and RX power consumption trends intersection can be explained by the succession of different roles the sensor node has during the communication process, because of the random nature of the mechanism used. When the random timer fires and the sensor node broadcasts to find a partner it takes the role of initiator in this communication process and TX power consumption is bigger than RX. On the contrary, if the sensor node has received a broadcast message, it unicasts to its neighbor, and TX energy consumption is lower than RX.

ISBN: 978-1-61804-355-9

Similarly, other simulations were performed for different size WSNs with 30, 50, 70 and 100 Tmode Sky sensor nodes. Results are listed in Table 5.

It can be seen a rather unexpected result regarding the convergence time. Fig.4 illustrates in the first subplot the relation between number of sensor nodes and convergence time. Evaluated convergence time value remains almost constant with the increasing of sensor nodes number. Moreover, instead of increasing one can see that it easily degreases as more sensing nodes seek to reach a common decision. Though, if the number of sensor nodes is increased more, convergence time eventually increases. This is because for small number of sensor nodes the communication mechanism can be slow, because most of the sensor nodes wait for their dialog turn. Different topologies and adjustment of the communication mechanism parameters may reduce convergence time for WSNs consisting of few sensor nodes, but this topic is not detailed in this paper. An increased density of sensor nodes ensure multiple available free for dialog neighbors.

Table 5 Convergence time, number of convergence steps and average power consumption

Nodes nb.	Time (s)	Steps	Power (W)
30	35	122	0.83
50	32	215	1.46
70	29	600	4.10
100	36	760	5.17

With regard to the convergence step number, it increases linearly together with the raise of sensor nodes, as one might expect. Relation between the number of sensor nodes and evaluated convergence steps is illustrated in the second subplot of Fig. 4.

As power consumption derives from the number of convergence steps and on line energy estimation mechanism it is reasonable that power consumption increases with the number of sensor nodes. Evaluated power for different number of sensor nodes is shown in the bottom subplot in Fig.4. If the power consumption is converted to watt-hour, an energy equivalent to 0.052Wh is obtained. Assuming that the sensor node is power supplied by an alkaline battery, which provides about 6Wh energy, it fallows that a sensor node in a WSN of 100 nodes is able to execute 120 cycles of binary consensus algorithm.

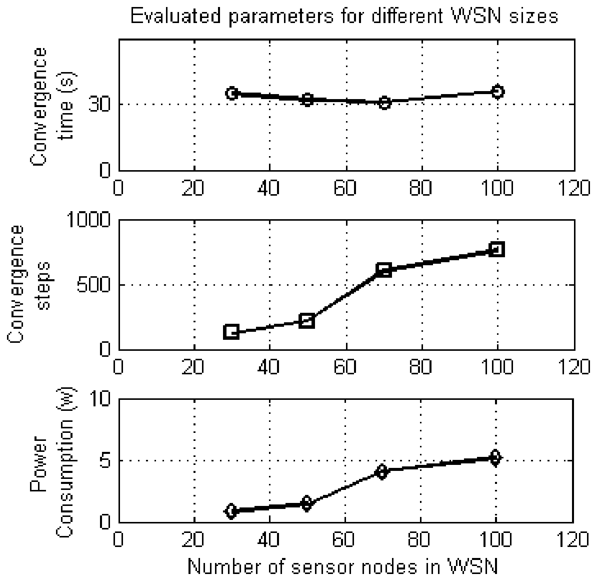

Fig.4 Evaluated parameters for different WSN sizes

5 Conclusions

Binary consensus plays a significant role in decentralized event detection systems based on wireless sensor networks. Main advantage of binary consensus algorithms comes from its limited complexity. This makes it accessible to implement on real sensor node platforms. Main challenge is to develop and implement a flexible and robust communication mechanism.

New available virtual tools for simulation and prototyping enable accessible development and great virtual test bed for researchers. Contiki offers a flexible WSN simulation environment for Contiki OS based sensor nodes, called Cooja.

The paper presented an adjusted binary consensus algorithm, tested on different size WSNs comprising Tmote Sky sensor nodes using the Contiki/Cooja Simulation Environment, by means of convergence time, steps and particularly power consumption. The power consumption evaluation required an online energy estimation mechanism implementation. Experimental results showed that time convergence can vary in a rather unexpected way for different number of sensor nodes. As expected, simulations showed that a large number of sensor nodes involves a big number of convergence steps. Results showed that power consumption is indeed linear dependent on the number of sensor nodes. Regarding WSNs with large number of sensor nodes, due to the estimated number of available cycles of binary consensus algorithm execution, a network architecture based on clusters of nodes should be considered.

A significant proportion of this work consists of a node-to-node communication mechanism implementation, in a random neighbor communication approach. Our main view is that this type of approach would easily apply to other consensus decision making algorithms.

Acknowledgement

The work has been funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Ministry of European Funds through the Financial Agreement POSDRU/187/1.5/S/155536 and partially by National Research Programme STAR, project 71/2013: Multisensory robotic system for aerial monitoring of critical infrastructure systems-MUROS.

References:

[1] Markus C. Beckera, Pasquale Salvatoreb, Francesco Zirpolib, The impact of virtual simulation tools on problem-solving and new product development organization, *Research Policy*, Volume 34, Issue 9, November 2005, pp. 1305–1321.

[2] A. Dunkels, N. Eriksson, F. Österlind, and N. Tsiftes, "The Contiki OS - The Operating System for the Internet of Things." [Online], at http://www.contiki-os.org/.

[3] Noor Al-Nakhala, Ryan Riley, Tarek Elfouly, Distributed algorithms in wireless sensor networks: An approach for applying binary consensus in real testbed, *Computer Networks*, 79, 2015, pp. 30-38.

[4] Achour Mostefaoui , Michel Raynal, Frédéric Tronel, From Binary Consensus to Multivalued Consensus in asynchronous message-passing systems, *Information Processing Letters Volume 73*, Issues 5–6, 31 March 2000, Pages 207–212.

[5] A. Gogolev, N. Marchenko, L. Marcenaro, C. Bettstetter, Distributed Binary Consensus in Networks with Disturbances, *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, Volume 10 Issue 3, October 2015, Article No. 19.

[6] Grigore Stamatescu, Iulia Stamatescu, Cristian Dragana, Dan Popescu, Large Scale Heterogeneous Monitoring System with Decentralized Sensor Fusion, *The 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Application 24-26 September 2015, Warsaw, Poland*

[7] M. Draief, M. Vojnovic, Convergence Speed of Binary Interval Consensus, *INFOCOM, 2010 Proceedings IEEE*, March 2010, pp. 1-9.

[8] Adam Dunkels, Joakim Eriksson, Niclas Finne, Nicolas Tsiftes, Powertrace: Network-level Power Profiling for Low-power Wireless Networks, *SICS Technical Report T2011:05*, March 2011, ISSN 1100-3154.

[9] Tmode Sky Datasheet, [Online] at http://www.eecs.harvard.edu/