Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-13 (2016) 259–264

The method of saving data integrity for decentralized network of group of UAV using quantized gossip algorithms

Kirill Tyushev * **Konstantin Amelin** *,** **Boris Andrievsky** *,***

* *St. Petersburg State University, St. Petersburg, Russia (e-mail: kirill.tyushev8@gmail.com, konstantinamelin@gmail.com, boris.andrievsky@gmail.com)*
** *Norwegian University of Science and Technology, Department of Engineering Cybernetics, Trondheim, Norway*
*** *Institute for Problems of Mechanical Engineering of RAS, Russia*

Abstract: In this work the method of communication and maintenance of integrity of data in the decentralized network of autonomous mobile robots is suggested. The software and hardware feasibility of such method is proposed. For collective storage and maintain integrity of data in decentralized group of mobile robots the algorithm of checksum calculations is proposed. Quantized gossip algorithms are considered as basis of checksum calculations algorithm. The local voting protocol is used for load balancing of mobile robots. For the programming implementation of the communication of data multi-agent technology is used. An ultra light, unmanned aerial vehicle (UAV) is considered as the mobile robot.

Keywords: Load balancing, UAV, consensus achievement, multi-agent networks, quantized gossip algorithms, mobile robots, data storage.

1. INTRODUCTION

The use of various types of mobile robots to perform tasks of monitoring and investigating areas is increasingly becoming a part of our daily lives Wulff et al. [2013], Apvrille et al. [2014], Samad et al. [2013] and etc. In recent years, there has been an increasing interest in groups of mobile robots, rather than isolated machines Amelin et al. [2013a,b], Yoshida et al. [2014], Sukop et al. [2014]. In most cases, during the implementation of group work, a centralized control scheme, in which there is a common leader and ground control center, is used. The main disadvantages of this approach are the need for constant communication of each member with the leader and the need for the leader to work in the group. To solve this problem a decentralized management system can be used.In recent year, the multi-agent systems is widely used as the solution of this task. Where is the main active elements are agents that have the same importance and functionality to the system Granichin et al. [2013], Van Der Hoek and Wooldridge [2008], Olfati-Saber et al. [2007].Management of a group of robots without a single center or common leader can be achieved via organizing a decentralized network by the use of multi-agent technology. This also gives the opportunity to build difference topology relationships between them. To execute decentralized network with varying topology, the task of hardware implementation must be resolved. Regarding this issue, communication technology must be developed. This includes developing hardware, software and algorithmic parts. One of the main tasks for autonomous mobile robots is to research of area. They need to collect, process and send data to other robots or users Bullo et al. [2009], Yu et al. [2010]. Group of mobile robots has a number of advantage compared to single robot. One of this is higher warranty of task execution due to distributed data collection and storage. Komarov et al. [2008], Chilwan et al. [2014], Granichin et al. [2015]. In case of lost part of the robots of group, for example, due to of damage, we can transmit and store all of collected data between all robots of group. This approach requires a lot of robots internal memory and high speed communication channel. In case of lost part of the robots of group, for example, due to of damage, the easiest way to preserve their integrity is to store a copy of all the data on the other robots on each mobile robot. However this will require a large amount of memory on each of the hard drives of the mobile robots. To avoid the storage of redundant data, RAID-like circuits can be used which can compute checksums and then restore data based on those results Plank et al. [1997]. A lot of attention was paid to obtain the corresponding consensus conditions for such systems (see e.g. Ren et al. [2007], Amelin et al. [2012], Amelina et al. [2012], Lewis et al. [2014], Granichin [2015], Amelina and Fradkov [2012]).

RAID (redundant array of independent disks) - an array of multiple storage devices interconnected by high performance communication and perceived external system as a whole. Such an array must maintain the integrity of the data in case of failure of one or more devices, as well as provide high-speed reading or writing data. Each storage

⋆ This work was supported by RFBR (projects 16-07-00890, 15-08-02640). The survey of the recent developments (Sections 4,5) is performed in IPME RAS under sole support of RSF, grant 14-29-00142.

10.1016/j.ifacol.2016.07.961

device can be regarded as a fully independent entity, being arbitrarily far from the other devices. In this context, RAID can be represented as a multi-agent system, wherein as a single agent serves one storage device. Moreover, the different agents can be located far from each other and connected by slow and unreliable communication channels, and data transmission is possible only at specific times. We call such storage RAID-like.

Example of RAID-like system is Ceph. Ceph is the distributed storage system which can calculate checksums according RAID-like scheme. But it calculates checksums by centralized and synchronous way. Such an approach could lead to poor performance with a large number of storage devices due to a large load on the central device. The new decentralized algorithm of checksums calculation was developed to solve such problems.

Relevance of new approach for checksums calculation consists of decentralized nature for data exchange between mobile robots. Current algorithms in RAID arrays (all of them based on Read-Solomon codes calculations Reed and Solomon [1960]) require one-time data communication from all robots to central computer which process this data to obtain checksums. If at any time we can not transmit data from some robot then in this time we can not obtain checksum. Such difficulties lead to very slow calculations.

In previous work Amelin et al. [2016] we calculate exact data checksums using local voting protocol. In such approach we need to do all calculations in real numbers. Such method requires high-performance hardware on mobile robots. In this work we use only integer numbers since the basis of a new method is quantized gossip algorithms Kashyap et al. [2007]. New approach can not calculate exact data checksums but it do not require high-performance processors on mobile robots.

This paper proposes a scheme of hardware implementation of communication in a decentralized network of mobile robots, based on an example of ultra-light UAV. The implementation of RAID-like schemes for decentralized computing of checksums on mobile robots is considered. It is shown that the checksum takes up less space in memory than all the data from all of the mobile robots, and that through this system you can recover data from all robots, even in the absence of communication or failure of members of the group. The local voting protocol is used for load balancing of mobile robots Amelina and Fradkov [2014] under noise and delay Granichin and Amelina [2015].

2. ALGORITHM FOR CHECKSUMS CALCULATION

Consider a multi-agent system of n agents. Let $i, i = 1, ..., n$ - agent number, t - time, $N = 1, ..., n$ - set of agents, $x^i_{t_k}, y^i_{t_k}$ - number on the agent under the number i at time t_k. We calculate and store m types of checksums of data on agents. In this case, the minimum number of agents, which preserves the integrity of the data is equal to $(n - m)$.

For simplicity, we assume that the data is transmitted without interference, without delay, and each agent stores only one number. Checksum calculation takes place in specified intervals. Fig. 1 T - time to compute checksums. Upon successful completion of all calculations on the current range in the future is possible to recover data that has been recorded in advance.

At the beginning of the interval is evaluated integral of f on the number on the agent, and the resulting value is stored in a pre-reserved area:

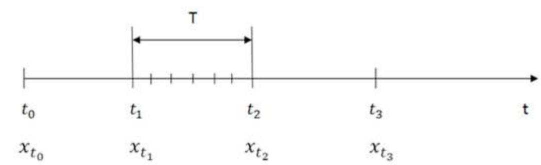

Fig. 1. Schema of checksums calculation

$$y^i_0 = f(x^i_{t_k}), i \in N \quad (1)$$

According to Kashyap et al. [2007] say edge $\{i, j\}$ is selected at time t, and let $D^{ij}_t = |y^i_t - y^j_t|$. Then, if $D^{ij}_t = 0$, we leave the values unchanged, $y^k_{t+1} = y^k_t$ for $k = i, j$. If $D^{ij}_t \geq 1$, we require that

$$\begin{cases} y^i_{t+1} + y^j_{t+1} = y^i_t + y^j_t, \\ if\ D^{ij}_t > 1\ then\ D^{ij}_{t+1} < D^{ij}_t,\ and \\ if\ D^{ij}_t = 1\ and\ (without\ loss\ of\ generality)\ y^i_t < y^j_t, \\ then\ y^i_{t+1} = y^j_t\ and\ y^j_{t+1} = y^i_t. \end{cases} \quad (2)$$

In this way, values on agents satisfy the following constraints:

- The value at each node is always an integer.
- The sum of values in the network does not change with time: $\sum_{j \in N} y^j_t = S$ for all t.

Let S be written as $NL + R$, where L and R are integers with $0 \leq 1R < N$. Vector y_t converges to quantized consensus distribution $y^i, i \in N$:

$$y^i \in \{L, L+1\}, i \in N, \sum_{j \in N} y^j_t = S \quad (3)$$

Thereby, on each agent we obtain estimated sum of values in the network which is equal LN or $(L + 1)N$.

Thus, we have learned to count estimated sum of the numbers y^j_0, where $j = 1, \ldots, |N|$.

When we change the function f we change the numbers y^j_0, where $j = 1, \ldots, |N|$. This allows to calculate different types checksums for numbers $x^i_{t_k}$, where $i = 1, \ldots, |N|$. Choose functions f and calculate checksums as follows:

$$\begin{cases} f(x^i_{t_k}) = 2^{ij} x^i_{t_k}, i \in N, j \in 1 \ldots m \\ m \leq |N|, x_{t_k} = (x^0_{t_k}, x^1_{t_k}, \ldots, x^n_{t_k}) \end{cases} \quad (4)$$

$$\begin{cases} h_1 = (2^0, 2^1, 2^2, \ldots, 2^{n-1}) \\ \ldots \\ h_m = (2^{m0}, 2^{m1}, 2^{m2}, \ldots, 2^{m(n-1)}) \end{cases} \quad (5)$$

$$\begin{cases} S_1 =< h_1, x_{t_1} > \\ \ldots \\ S_m =< h_m, x_{t_k} > \end{cases} \quad (6)$$