DOI: http://doi.org/10.25728/cs.2022.3.5

THE SIMULTANEOUS START OF ACTIONS IN A DISTRIBUTED GROUP OF AUTOMATIC DEVICES: A DECENTRALIZED CONTROL METHOD WITH A SIGNAL REPEATER

G.G. Stetsyura

Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia

✉ gstetsura@mail.ru

Abstract. This paper proposes a method for accelerating decentralized synchronization processes in the distributed control of a group of stationary or mobile automatic objects. With this method, the objects pass to specified states or affect the environment simultaneously or with required time delays. Some examples of such objects include actuators, computers in a computing cluster, distributed data processing facilities in supercomputers, and mobile robots. The object's action depends on the current state of all objects and the environment. The actions should start with minimum delay after detecting the possibility to perform them. Arbitrarily located sources of executive commands and their receivers are synchronized by exchanging signals and messages between objects through an intermediary (a signal repeater). Means are used to accurately measure the time intervals of signal transfer between each object and the repeater. Group operations are used to accelerate synchronization processes. These operations involve a large number of objects simultaneously. The object's data are used in operations simultaneously. Data are processed during their transmission without extra time. Operations are executed by network devices of the system objects and the common network device without any computing facilities (the repeater).

Keywords: simultaneous start of group operations, decentralized control, synchronization of mobile objects, fast distributed intranet computing, multilayer synchronization.

INTRODUCTION

In this paper, we accelerate the decentralized control of starting joint actions in a distributed group of digital objects: computers in a computing cluster, distributed data processing facilities in supercomputers, mobile robots, and actuators affecting an environment.

The problem statement is as follows. Consider a distributed group of sources that jointly create a common command and send it to a distributed group of receivers (command executors). The objects in the groups have an arbitrary arrangement, can change their location, and communicate through a network. Having a command, all receivers must perform the corresponding actions simultaneously or with the time delays specified for each receiver in the command. The time instant of command sending is unknown in advance and depends on the current state of all objects and the environment. The objects should start the actions with the minimum possible delay after the sources send the parts of the command. When objects operate in a distributed control system, it is also important to reduce the delay with which system objects form a common command. All objects, sources and receivers, must act without centralized control.

The solution of this problem consists of two parts: for the sources and receivers of the command, respectively. The actions of sources were presented in the author's previous publications and are briefly described in Section 3 below. We introduce an additional communication link for objects, the signal repeater *RS*. This is a simple device without computing facilities and even logical elements. Receiving signals of objects at some frequencies, *RS* just retransmits them to objects at other frequencies. It cannot generate commands and therefore does not serve as a control center. Receivers have signals only from *RS*. Due to the use of *RS,* we establish the following key results.

With the proposed synchronization of object actions, sources send messages to *RS* with the simultaneous arrival of the same-name binary digits of all messages. For receivers, *RS* acts as a single source replacing the previous group of sources. Now objects must consider only changes in their distances to *RS*. Sources send messages to a single receiver (*RS*). Receivers have messages from *RS* only. This approach simplifies the network facilities and reduces the command execution time. By adding *RS,* we eliminate interference from source signals coming to the receivers. Without *RS,* signals from a group of sources, even sent simultaneously, would arrive at the receivers as interference at different, almost uncontrollable, time instants.

This paper considers group operations and commands executing distributed control and computational operations in a time independent of the number of objects (the simultaneous participants of the operation). As shown below, distributed object computers execute group operations at high speeds due to *RS*.

To solve this problem, it is necessary to determine the transfer time of signals between objects and *RS*. Many solutions have been developed in different technical fields. For this paper, the most useful results are two standardized solutions for the clock synchronization of distributed objects and high-precision measurement of distances between objects. The IEEE 1588-2008 Precise Time Protocol (PTP) [1] is widely used in the industry. In PTP, signal transfer times are measured to synchronize the clocks of objects. Depending on the application, the accuracy varies from tens of microseconds to eight nanoseconds. Within the White Rabbit (WR) project [2, 3], picosecond methods were developed for measuring signal transfer times between objects in accurate physical experiments at CERN. The IEEE 1588-2019 High Accuracy Default PTP Profile (HA) [4] is a novel standard combining both solutions. In PTP and WR, objects interact in the master-slave mode. Both PTP and WR can operate on large networks.

PTP and WR solutions are applied below with some modifications due to the problem statement. This paper considers intensive data exchange between objects to control their actions and perform distributed computations, requiring small delays in data delivery. Therefore, the matter concerns only the systems in which the distances between objects vary from fractions of a meter to several hundred meters.

The solutions proposed below are oriented to tasks with the unpredictable execution time of commands by receivers, so the clock is not applied. There is no master object, and all objects have equal rights.

The presence of *RS*, an additional device between objects, raises the natural question of reducing the fault tolerance of the system. Due to the simplicity of *RS*, the number of such additional devices can be increased to replace the failed ones. In addition, the *RS* functions can be transferred to any object. This paper does not discuss the issues of fault tolerance.

At the beginning of the Introduction, we have mentioned different digital objects to apply the methods. However, many technicalities need to be specified for a particular object. For example, many mobile objects require communication only by exchanging non-directional radio signals through a single *RS*. In a supercomputer, it is reasonable to apply directional optical communications with signal switching through thousands of simultaneously accessible *RS*s. In each of these cases, the methods described below remain the same.

This paper is organized as follows. Section 1 outlines the principles of time measurement in PTP and WR useful for further considerations. The communication structure to synchronize the actions of sources and receivers of commands is described in Section 2. Section 3 presents a method for receivers to execute source commands simultaneously or with the time delays specified in the command for each receiver. In Section 4, we introduce a distributed control method for source interactions. Section 5 considers the multilayer synchronization of command execution by receivers. Having executed a command, the receivers of a layer become command sources for the next layer of receivers. In Section 6, a summary of group operations is given. Section 7 shows the connection between the network operations proposed in the paper and the operations of associative computing facilities.

1. TIME INTERVAL MEASUREMENT IN PTP AND WR

The basic scheme for measuring time intervals in PTP is shown in Fig. 1*a*.

Here two objects interact, conventionally termed a master and a slave. At a time instant t_1, the master sends to the slave a signal to start synchronization and its clock time. This message arrives at a time instant t_2. At a time instant t_3, the slave sends the master a reply

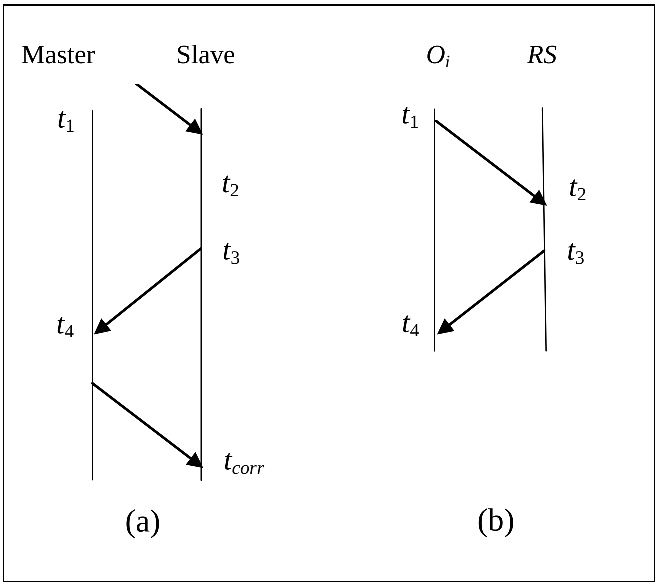

Fig. 1. Time measurement in PTP: the basic scheme and its simplified version.

signal and its new clock time. At a time instant t_4, the master receives the slave's reply, determines the distance between this pair of objects, determines the distance to the slave, and reports it to the slave. The slave corrects the clock time. Except for several important details, this is the basic time correction scheme in PTP.

For this paper, we need a simplified version of the measurement scheme (Fig. 1*b*) without clock but with a signal repeater (*RS*). For optical signals, a passive retroreflector can be used as *RS*. An arbitrary object O_i launches its timer, sending a signal to *RS* at a time instant t_1. The signal arrives at the *RS* at a time instant t_2. After the triggering delay of *RS*, the signal is sent to *RS* at a time instant t_3. At a time instant t_4, the object O_i determines the signal transfer time between O_i and *RS* as $T_{OiRS} = t_4 - t_1 - (t_3 - t_2)$.

WR uses a more accurate phase method for measuring time intervals. In [5], a simple electronic device was proposed to measure the signal transfer time between objects. The femtosecond accuracy was achieved. Below, this method can be applied directly or to control the stability of all objects participating in the measurement process.

2. COMMUNICATIONS BETWEEN COMMAND SOURCES AND RECEIVERS

Figure 2 shows the structure of communications to exchange signals and messages between command sources and receivers.

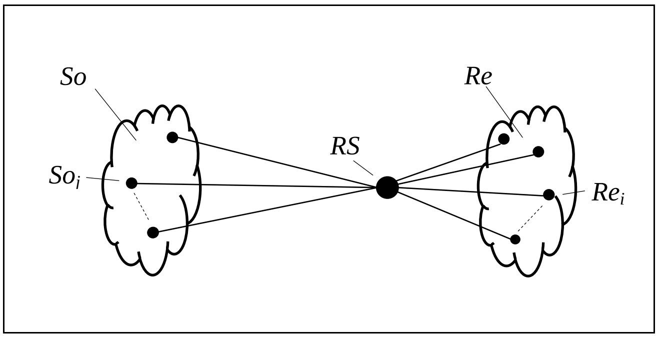

Fig. 2. Communications between command sources and receivers.

The object system includes a group *Re* of command receivers $\{Re_i\}$, a group *So* of command sources $\{So_j\}$, and a repeater *RS* of signals coming from sources and receivers. Up to Section 4, we assume that only one arbitrary source from *So* sends a command to receivers from *Re*. The source sends the command signal to *Re* indirectly (through *RS*). The bit “1” of the command code is sent to *RS* on a carrier frequency f_1; the bit “0,” on another frequency f_0. The repeater translates these signals into other signals $^*f_1$ and $^*f_0$, respectively, and sends them to receivers. The signals do not change during WR operations. As noted in the Introduction, objects receive only the signals transmitted by *RS*.

Therefore, *RS* is a simple device without logical elements. It does not actively participate in the control of objects’ actions.

RS underlies the synchronization approach proposed in this paper. In addition, *RS* reduces the amount of transferred data and time required for Re_i to organize the synchronous execution of the command. As noted in the Introduction, the problem solution will be time-consuming without collecting messages in a single point (*RS*). These features are discussed in detail below.

Communications between objects can be wireless or wired. Wireless communications offer additional capabilities. For example, only wireless communications are acceptable for mobile objects. In stationary supercomputers, wireless communications between a group of objects and a group of *RS*s allow quickly reconfiguring the system when solving a single task. They also simplify system recovery in case of *RS* failures. They eliminate the duplication and triplication of repeaters. For example, if there is at least one redundant repeater, objects will switch to it without intermediate switches in case of failure. A failure can be detected due to the broadcast transmission of the logical scale accessible to all objects simultaneously (see Section 4).

3. SYNCHRONIZED COMMAND EXECUTION BY RECEIVERS

This section presents an accurate synchronization method for receivers (command executors) with the communication scheme in Fig. 2. Before receiving an executive command to start the synchronized execution of the corresponding actions, each receiver gets a description of its action in a preliminary command. Such commands can be sent by different sources in random order or as a single message consisting of messages from individual sources. This method is discussed in Section 4.

Before implementing the executive command, each receiver must measure the transfer time of the signal (optical or radio signals having the speed of light c) between the receiver and *RS* (the distance to *RS*). We propose a procedure with additional frequency channels in which measurements can be performed independently and simultaneously with other communications between objects.

The first task of the receivers is to launch the time without involving a special control center, which sets the order of measurement for them. For this purpose, the receivers from *Re* need to determine the distance to *RS*, and they send a special synchronization signal S to *RS*. The signal duration must be at least T (the greatest signal transfer time between *RS* and any receiver). A receiver sends the signal S only when not receiving the signal S sent by other sources and returned from *RS*. If the signal duration is not smaller than T, the individual signals S of different objects are superimposed in the common signal S of variable duration. The repeater converts S into a single signal S_{rs} and transmits it to receivers *Re*.

The time instant when the signal S_{rs} is completely received is treated by the receiver as the signal ${}^{*}S_{rs}$ to start synchronization. This signal is created without any control center; also, see the paper [6]. Let the receivers from *Re* be numbered. The object Re_i with the smallest known number i (e.g., $i = 1$) measures its distance to *RS*. All receivers do the same. This process can be performed continuously and simultaneously with other interaction processes of the objects. Some versions of this action were considered in the paper [6]. In particular, suppose that high synchronization accuracy is not required (the time intervals of a duration below T are indistinguishable). Then the objects may exchange signals directly without *RS*.

Upon measuring their distances T_i to *RS*, the receivers Re_i will start the synchronous execution of the source command. To do this, each receiver follows several steps described below.

Step 1. The receiver Re_i calculates the delay $d_i = C - T_i + a_i$. Here $C \geq T$, and $a_i \geq 0$ is an additional time delay for Re_i (possibly zero). The value $C \geq 0$ is used to consider the time cost of additional object actions.

Step 2. Upon receiving a command, each Re_i will perform the command actions with the delay d_i.

Leaving *RS*, the command will arrive at Re_i after the time T_i. Hence, given the delay d_i, any Re_i performs the action at the time instant $\tau = T_i + C - T_i + a_i = C + a_i$. All Re_i will perform the command simultaneously at the time instant C after the command leaves *RS* or with the delays a_i, as required.

Thus, after the command leaves *RS*, all objects will pass to synchronous execution at the time instant of its delivery to the farthest object from *RS*. If all objects know $T_{\max}$ (the signal transfer time between *RS* and the farthest object), and it is acceptable to replace C with $T_{\max}$, the transition to the synchronous state will occur in the minimum possible time.

Now we determine the time instant of measuring T_i by receivers Re_i. Recall that the measurement procedure is continuous. Hence, we can select the last measurement before receiving the command by the object. But it is possible to take the measurement when the receiver gets the command. If all Re_i know the time for determining the values T_i and d_i, then it suffices to set an upper bound on this time for all Re_i so that the receivers correct possible time variations.

The accuracy of measuring T_i can be significantly improved by the following method, oriented to tough time requirements. (They are necessary to coordinate computers in receivers Re_i.) In this method, Re_i will also measure the distance to *RS* after the other receivers complete their measurements. But all these measurements are performed within one common message for all receivers. In this message, each Re_i has a very short time interval Δt for the measurement.

We make several assumptions. First, before applying the proposed method, receivers Re_i determine the time intervals T_i and calculated the values d_i. Second, the constant C is given; in the best case, $C = T$ [6]. Third, each Re_i is allocated a time interval Δt_i of the same duration Δt to recalculate T_i. The method includes the following steps.

Step 1. As described above, receivers Re_i send the signal S to *RS* and receive the signal S_{rs} back. By the signal ${}^{*}S_{rs}$ (the completion of S_{rs}), each Re_i sends to *RS* the test signal δt of a duration smaller than Δt, placing δt in the center of Δt_i. Note that each Re_i does this alternately on i within its interval Δt_i.

Step 2. Since all Re_i know the value Δt, they will send their Δt_i in one common message SC of duration $n\Delta t$, where n is the number of receivers.

Step 3. The signals δt of the message SC arrive at RS. Here, they are converted to $^*\delta t$ and then returned to receivers. Receiver Re_i determines the new value T_i by the shift δt within Δt_i.

The method is effective under two conditions. First, the signal δt must not leave Δt_i for any displacement of Re_i. Second, the duration $n\Delta t$ must be smaller than that of the previous method for determining T_i (with a group of individual messages). Let us demonstrate that both conditions hold.

For stationary Re_i without any external impacts, the signal δt is in the center of Δt for any T_i. Otherwise, δt is shifted.

Let L be the maximum distance between RS and any Re_i, and v be the maximum speed of Re_i. Then during the measurement time T_i, the receiver will move at the distance $^*L = T_i v$. The signal δt will shift within Δt by no more than the time interval $^*\Delta t = {}^*L/c = T_i v/c$. The interval Δt must be at least $2^*\Delta t$ for δt not to move to the adjacent interval Δt. The entire measurement cycle for n receivers will be done in the time $2nT_i v/c << nT_i$. Allocating a separate frequency channel for each Re_i will eliminate the dependence on n and reduce the admissible time of measuring the distance to RS for the entire group Re.

The solutions described above ensure minimum delay when executing the command sent to the receivers. Indeed, RS (if exists) acts as the only source of the command. The receiver measures the distance to RS before getting the command, and there is no delay at the beginning of its execution. Next, each receiver has to start executing the command only when all receivers get it, delaying the execution due to its distance to RS. As shown in this section, the receivers have all the necessary information for this in advance; upon getting the command, the receiver starts its execution with the corresponding time shift. Thus, the presence of RS eliminates command execution delays. Delays inevitably occur without RS.

If a group of sources forms a common command without RS, then their simultaneous actions are achieved only by replacing RS with one of the sources (the leader). The leader will act as RS. Such a process takes unacceptably much time.

In Section 4, sources send messages to receivers only via RS, adjusting the message arrival at RS depending on their distance to RS. Acting like receivers, the sources determine the distance to RS and then send messages so that they reach RS at the same time or in a given order. The sources determine the distance to the RS in advance, and a common command will be formed without delay, e.g., using the solutions of Section 6.

The method for measuring signal transfer (PTP, WR) without changing the paper's solutions can be replaced by another known method. Thus, in all cases, the presence of RS minimizes the delay when executing a common action of all receivers.

We have described the synchronized execution of commands in the case of a single RS sending signals to all receivers. Section 5 considers a more general problem. But first, in Section 4, we discuss the joint synchronous actions of a group of sources.

4. SYNCHRONOUS ACTIONS IN A GROUP OF SOURCES

The synchronous actions of a group of sources were considered in [6]. They are used repeatedly in this paper. A summary of such actions that corresponds to Section 3 is given below.

Similar to the actions of receivers, in Section 3 the sources organize their actions as follows. Sources So_j from the group So, ordered by j, send the signal S to RS. In response, RS sends the signal S_{rs} and the signal $^*S_{rs}$ (the sign of completing S). Then sources So_j alternately determine the distance to RS and calculate the delays $D_j = C - t_j + a_j$, by analogy with d_i , C, T_i , and a_i from Section 3. Now the sources can synchronously send messages to RS (and commands to receivers) without any control center.

To describe further actions of the sources, we introduce a logic scale LS, i.e., a sequence of bits equal to the number of sources in So. Let source So_j need to transmit a message to RS. This source enters one into the jth bit of the scale LS and transmits it to RS using the signal f_1. The other bits of LS may not contain binary values, or So_j enters zero into them and transmits it using the signal f_0. In the second case, several signals f_1 and f_0 arrive at RS (and further at Re_i) simultaneously from different sources. They are perceived as a pair of signals in some operations of Re_i but as the signal f_1 in most operations.

To start interacting with RS, the sources send to RS the signal S and gent the signals S_{rs} and $^*S_{rs}$ in response. After that, the sources send their scales LS to RS with the delays D_j and get a combined scale $^*LS$, with the same-name bits of all scales received by RS. Now So_j can send their messages to RS in a given order without delays on the sources not requesting message transmission.

Thus, we have obtained a useful result. The sources order their messages by sending them to RS. Now RS acts as a single source, sending messages-commands from RS to receivers. At the same time, sources can be receivers to coordinate joint actions. In particular, So_j can acknowledge the agreement of the entire group So to execute a command sent to Re. As shown in Section 6, besides signal repeat, RS can also execute several operations distributed among the members of *So* without logical elements.

In contrast to the synchronous actions of objects, there is no possibility to specify the occupation time of *RS* for the asynchronous actions of objects. In this case, we apply barrier synchronization [7]. When executing a common operation, one or more objects send a signal *B* that is accessible to all objects and differs from all other signals. When some object completes its work within the common operation, it stops transmitting the signal *B*. When all objects do so, the transmission of the signal *B* stops. In the absence of this signal, other objects can start the next operation. Without these actions, the value *C* will be chosen unreasonably large for the asynchronous operations of objects. The operations of sources (Section 4) are used in the next section for the multilayer synchronization of actions in *Re*.

5. MULTILAYER SYNCHRONIZATION FOR GROUPS OF RECEIVERS

In contrast to Section 3, here the group of objects is divided into subgroups (layers) Le_1, Le_2, ..., Le_n (Fig. 3). Within a layer, objects act as discussed in the previous sections. The signals they exchange are inaccessible to the objects of other layers. But the repeaters *RS* of the layer controlled by layer objects can merge with the repeaters *RS* of the next layer. The first layer of receivers gets source commands directly from *So* of the layer (Section 3). Then the group *Re* of first-layer receivers through its repeater acts as a group of sources for second-layer receivers, coordinating their actions as described in Section 4. For this purpose, the repeaters *RS* of these layers act jointly: the second-layer repeater transmits first-layer signals to second-layer objects and sends second-layer signals to the first layer. Next-layer receivers act in the same way. In the simplest case, last-layer receivers need to act synchronously with respect to external objects. In the more complex case, the computers of receivers and computers of sources in a layer exchange messages through their *RS*s, which are simultaneously accessible to all these objects and should be used to correct their actions. Hence, additional time is required. Let us start with the simplest process (process 1).

Process 1. The objects are divided into *k* layers, $k = 0, ..., n$. The receivers of layer *k* interact with the receivers of layer $k + 1$ by switching to the source mode. The new sources of layer *k* act as described in Section 3, synchronizing the receivers of layer $k + 1$.

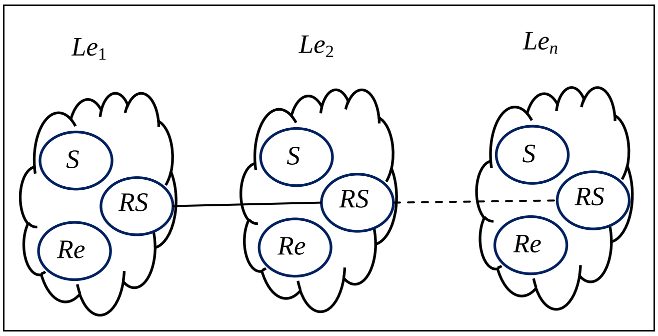

Fig. 3. Multilayer synchronization of groups of receivers.

Moving from layer to layer, this process reaches the layer whose receivers are to be synchronized. The synchronization will be done. The intermediate layers are forbidden to perform any external actions.

If the receivers of all layers are required to perform external actions simultaneously without *RS* with the same time *C* for all layers, then we use process 2. (It complements the actions of process 1.)

Process 2. Upon getting a command, the receivers of arbitrary layer $k \leq n$ repeatedly calculate the value $F = n - k$, increasing *k* by one. In each iteration, the value *F* is calculated with the delay *C*.

If $F = 0$, the receivers of the layer will perform an external action. In the first iteration (for layer $k = 0$), we obtain $F = n$; for layer $k = 1$, $F = n - 1$, and so on. As a result, the receivers in all layers will get $F = 0$ when the last layer receives the command. They will simultaneously perform the external action.

Besides external actions, the computers of layer *k* objects may need data and command exchange as well as distributed computations. For this purpose, they use the common resource RS_k. Suppose that an upper bound $^*C$ can be specified for the occupation time of this RS_k by layer objects. Then the occupation time is considered by replacing *C* with $^*C$ if the layers work alternately.

In the presence of asynchronously acting objects, the following two-phase synchronization process yields the best result.

The synchronization preparation phase. In this phase, the data are asynchronously prepared for transmission to the receivers by first-layer sources. All actions are performed using barrier synchronization (Section 4). If necessary, first-layer receivers also asynchronously form additional data to prepare actions as second-layer sources. The transition to second-layer actions is performed by the barrier synchronization signal. All layers perform their work alternately. All layers do not need to perform external actions simultaneously. (This is the aim of the next phase.)

The synchronization phase. The process is similar to process 2, but all layers are renumbered in the reverse (the last layer n has number 0). The objects of the layer with the new number $k = 0$ act as sources and start synchronizing the receivers of all layers. Upon getting the command, the objects repeatedly calculate the value[*] $F = n - k$, increasing k by one, with the delay $^{**}C$. When $F = 0$, the receivers in all layers simultaneously perform the external action.

The synchronization phase is faster: the objects just pass the command to the previous layer. Therefore, $^{**}C < C$. In the synchronization phase, a feedback link is additionally introduced to pass the results of computations of the next layers to the previous ones.

Thus, we have obtained the following result. The objects synchronize their actions and perform them simultaneously in all layers of the multilayer structure without any control center.

6. A SUMMARY OF THE GROUP OPERATIONS PERFORMED ON THE NETWORK

We summarize the group operations considered in the paper as a means of group interaction of objects. Several operations were developed for this paper; others were introduced by the author earlier and adapted for the purposes of this paper. The operations listed below were developed at different times at Trapeznikov Institute of Control Sciences, the Russian Academy of Sciences (ICS RAS). The following text thoroughly describes the basic principles of group operations without addressing the primary sources.

The main properties of operations are as follows. Input data for an operation come simultaneously from a group of distributed objects (data storage devices). These data are processed simultaneously during their transmission without increasing the data transfer time. Operations are performed in network facilities of the object system without using computers and other computing devices. The time to obtain the result does not depend on the amount of data processed.

These results are a consequence of coordinating object actions through message synchronization processes. Regardless of the current arrangement and location of objects, their actions are synchronized by allocating a special object (signal repeater *RS*) for the groups. A group of messages is synchronously delivered to one object *RS* and subsequently forwarded to a group of receivers as a single common message for all sources. This is performed quite simply and quickly. But this message arrives at a group of randomly located receivers at different times. Using *RS*, the receivers introduce appropriate delays and simultaneously perform the required group action. According to the previous sections of the paper, the above properties hold for all **object action control operations**: the synchronization of sources, the synchronization of receivers, multilayer synchronization, the elimination of access conflicts, and barrier synchronization.

In addition, we list distributed **computational operations**: search for maximum and minimum, the bitwise logical AND and OR operations, and the analog-digital operations of counting, addition, and subtraction. These operations are not used here, but they accelerate system control when searching for objects with given properties and estimating the system state. They were described, e.g., in the paper [7]. Let us outline their operating principle.

To determine the maximum, objects send numbers in the binary representation, transmitting the highest digit at the first step. The next digit is transmitted only by the objects that transmitted one before, and so on. Hence, the maximum of the transmitted numbers remains. Replacing ones with zeros determines the minimum. It is possible to switch from the binary representation to other number systems if the digits of a number are represented by a scale in which all digits contain zeros, except for the digit corresponding to the digit value.

The bitwise AND and OR operations allow quickly estimating the state of all system objects. For this purpose, the object state is described by a scale, i.e., a sequence of binary digits equal to one if the corresponding attribute is present and zero otherwise. Ones and zeros are transmitted on the signal frequencies f_1 and f_0, respectively. The state of all objects is estimated with the simultaneous transmission of the scales to *RS* and combining same-name digits of object sequences in *RS*. When the AND operation is performed, the presence of f_0 in the scale digit of *RS* means that at least one object does not have the corresponding feature. Otherwise, the feature is inherent in all objects. For the OR operation under the same conditions, the presence of f_1 means that at least one object has the corresponding feature. Otherwise, the feature is absent in all objects.

To perform analog-to-digital operations, *RS* contains an analog-to-digital converter (ADC). Let the objects characterize their state with a scale (a sequence of attributes represented by zeros and ones). If objects send their scales to *RS* with combining same-name digits, then ADC will estimate the total energy of received signals and yield a value reflecting the presence of an attribute for all objects. This value will be received simultaneously by all objects. Combining such operations with the bitwise AND and OR operations allows estimating the state of the object system more accurately. The addition and subtraction of numbers in a non-binary number system were described in [7]. In these operations, the digits of numbers are represented by scales with one only in the digit of the scale corresponding to the digit value. The result also does not depend on the number of objects participating in the operation.

ADC operations require optical signal sources with stable energy. Modern technology allows obtaining an accurate digital value at the simultaneous summation of several thousands of signals. In [8], a simple LED source with an output power stability below 50 ppm/ºC was presented.

7. THE SOLUTIONS PROPOSED IN THIS PAPER: GENERAL ANALYSIS

The main results of this paper have been obtained with distributed group operations not used in known network systems. Their counterpart is associative operations proposed in the 1960s for concentrated associative computers. The associative operations are performed according to the following enlarged scheme.

In associate computers, the control center simultaneously sends a common command to a group of associative computer nodes with associative memory. The nodes perform the required actions. Their result is accessible to other nodes. It simultaneously comes to the center, which forms the next command based on the received results, and so on. The main application of associative computers was hard real-time control systems. An example is the STARAN associative supercomputer, which controlled aircraft movements at the J.F. Kennedy International Airport (New York, the USA). The group operations described in the paper can be considered a variant of associative operations with the following peculiarities. The operations are distributed and are executed directly in a simple network facility (a repeater). The repeater has no computing facilities; nevertheless, it executes all these group operations and serves for managing object interaction and simultaneously estimating the state of all objects. All types of object interaction described above are performed in a decentralized way without any control center. Thus, the structures proposed in the paper have both network and associative capabilities.

Note that at present, many researchers study networks executing message transfer and other functions (data transport management and distributed computing [9–12]). However, these important R&D works differ from those carried out in ICS RAS: new functions are performed by computers of network facilities, whereas group operations are not used.

CONCLUSIONS

This paper has proposed a decentralized control method for the simultaneous start of actions in a distributed group of stationary or mobile automatic objects. With this method, the objects start their actions with the minimum delay at an a priori unknown action instant.

The simultaneous start of actions in a group of objects with an a priori known action instant is successfully implemented in the IEEE 1588-2019 standard. Thus, the scope of this standard has been extended to real-time control systems by adding to the capabilities to start actions at an a priori unknown instant.

Due to network group operations, the proposed method has a fast response to emerging events.

REFERENCES

1. IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, *IEEE Std 1588-2008* (*Revision of IEEE Std 1588-2002*), 2008, pp. 1–269. DOI: 10.1109/IEEESTD.2008.4579760.
2. IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, *IEEE Std 1588-2019* (*Revision of IEEE Std 1588-2008*), 2020, pp. 1–499. DOI: 10.1109/IEEESTD.2020.9120376.
3. Girela-López, F., López-Jiménez, J., Jiménez-López, M., et al., IEEE 1588 High Accuracy Default Profile: Applications and Challenges, *IEEE Access*, 2020, vol. 8, pp. 45211–45220.
4. Sliwczynski, Ł., Krehlik, P., Buczek, Ł., and Schnatz, H., Picoseconds-Accurate Fiber-Optic Time Transfer with Relative Stabilization of Lasers Wavelengths, *Journal of Lightwave Technology*, 2020, vol. 38, no. 18, pp. 5056–5063.
5. Moreira, P., Timing Signals and Radio Frequency Distribution Using Ethernet Networks for High Energy Physics Applications, *PhD Thesis*, University College of London, 2014. https://discovery.ucl.ac.uk/id/eprint/1461109/1/PMmoreira_PhD_Final-signed[1].pdf.
6. Stetsyura, G.G., Decentralized Autonomic Synchronization of Interaction Processes of Mobile Objects, *Control Sciences*, 2020, no. 6, pp. 47–56. (In Russian.)
7. Stetsyura, G.G., Network Information-Computing Support of Automatic Mobile Objects Interaction, *Automation and Remote Control*, 2019, vol. 80, no. 6, pp. 1134–1147. DOI: 10.1134/S0005117919060110.
8. Bosiljevac, M., Babić, D., and Sipus, Z., Temperature-Stable LED-Based Light Source without Temperature Control, *Proceedings of SPIE OPTO*, San Francisco, USA, 2016, vol. 9754, pp. 1–6. DOI: 10.1117/12.2211576.
9. Tennenhouse, D.L., Towards an Active Network Architecture, *SIGCOMM Comput. Commun. Rev.*, 1996, vol. 26, no. 2. http://ccr.sigcomm.org/archive/1996/apr96/ccr-9604-tennenhouse.pdf.
10. Zilberman, N., Watts, P.M., Rotsos, C., and Moore, A.W., Reconfigurable Network Systems and Software-Defined Networking, *Proc. of the IEEE*, 2015, vol. 103, no. 7, pp. 1102–1124.
11. In-Network Computing, ACM SIGARCH, 2019. https://www.sigarch.org/in-network-computing-draft/

12. Kim, D., Towards Elastic and Resilient In-Network Computing, *PhD Thesis*, Carnegie Mellon University, 2021. http://reports-archive.adm.cs.cmu.edu/anon/2021/CMU-CS-21-143.pdf.

This paper was recommended for publication by V.G. Lebedev, a member of the Editorial Board.

Received November 28, 2021,
and revised May 5, 2022.
Accepted May 19, 2022

Author information

Stetsyura, Gennady Georgievich. Dr. Sci. (Eng.), Chief Researcher, Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ gstetsura@mail.ru

Cite this paper

Stetsyura, G.G. The Simultaneous Start of Actions in a Distributed Group of Automatic Devices: A Decentralized Control Method with a Signal Repeater, *Control Sciences* **3,** 46–54 (2022). http://doi.org/10.25728/cs.2022.3.5

Original Russian Text © Stetsyura, G.G., 2022, published in Problemy Upravleniya, 2022, no. 3, pp. 55–64.

Translated into English by *Alexander Yu. Mazurov,*
Cand. Sci. (Phys.–Math.),
Trapeznikov Institute of Control Sciences,
Russian Academy of Sciences, Moscow, Russia
✉ alexander.mazurov08@gmail.com